

University of Warwick institutional repository: <http://go.warwick.ac.uk/wrap>

A Thesis Submitted for the Degree of PhD at the University of Warwick

<http://go.warwick.ac.uk/wrap/2615>

This thesis is made available online and is protected by original copyright.

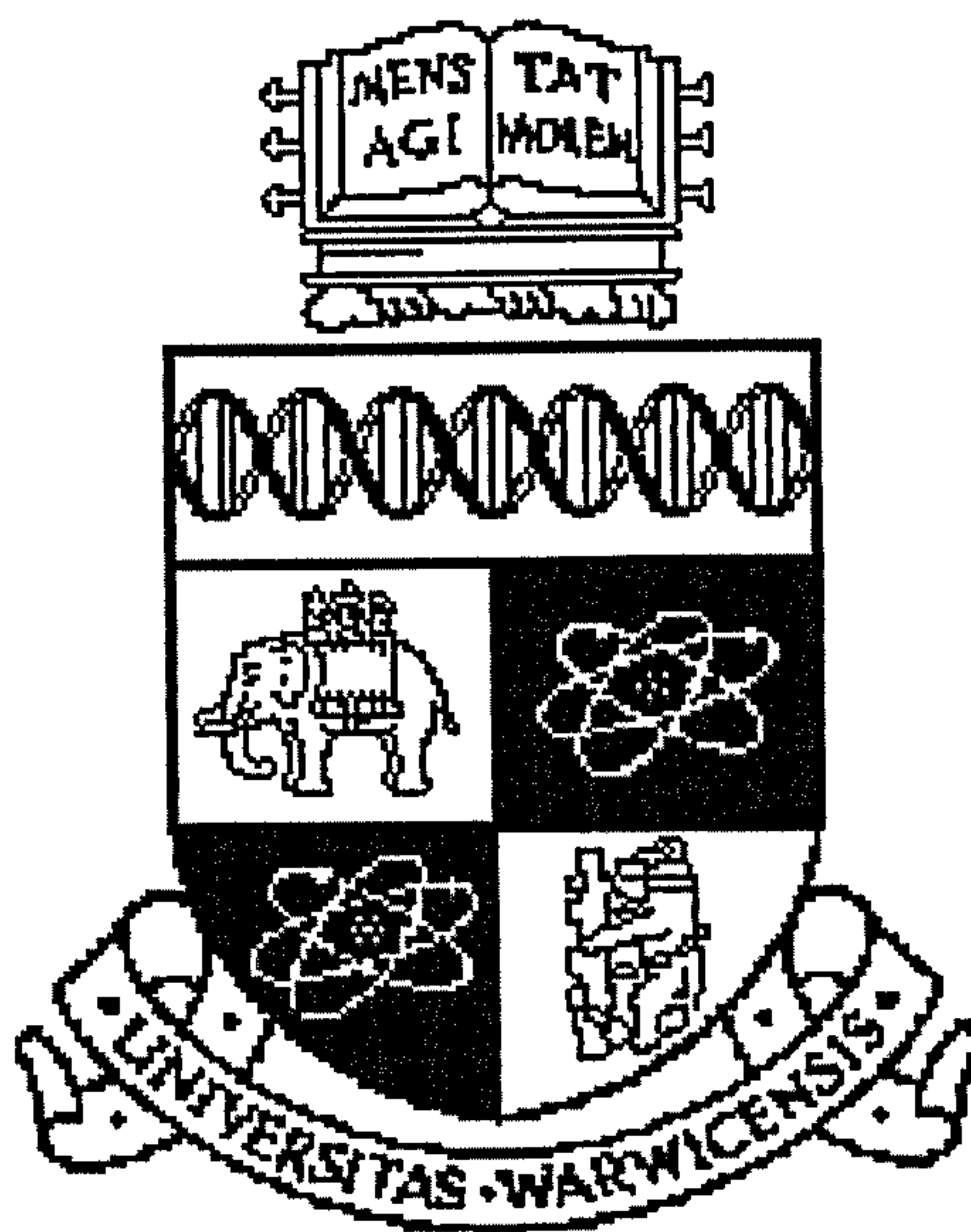
Please scroll down to view the document itself.

Please refer to the repository record for this item for information to help you to cite it. Our policy information is available from the repository home page.

A control strategy for Promoting Shop-floor Stability

by

You Sen Huang



Department of Engineering
University of Warwick
Coventry CV4 7AL
The United Kingdom

A thesis for the degree of
DOCTOR OF PHILOSOPHY

November, 1996

Abstract

This research aimed to study real-time shop floor control problem in a manufacturing environment with dual resource (machine and labour), under impact of machine breakdowns. In this study, a multiperspective (order and resource perspectives) control strategy is proposed to improve effectiveness of dispatching procedure for promoting shop floor stability. In this control strategy, both order and resource related factors have been taken into account according to information on direct upstream and succeeding workcentres. A simulated manufacturing environment has been developed as a platform for testing and analysing performances of the proposed control strategy. A series of experiments have been carried out in a variety of system settings and conditions in the simulated manufacturing environment. The experiments have shown that the proposed control strategy outperformed the ODD (Earliest Operation Due Date) rule in hostile environments, which have been described by high level of shop load and/or high intensity of machine breakdowns. In hostile environments, the proposed control strategy has given best performance when overtime was not used, and given promising results in reduction of overtime cost when overtime was used to compensate for capacity loss. Further direction of research is also suggested.

Acknowledgement

The research was supported by Warwick Manufacturing Group, and I like to acknowledge gratefully its help in making my study possible.

I thank Mr. Roy for his support and guidance throughout this research and the Ph.D programme. His encouragement gave me confidence to pursue this research. With his support, I was able to sustain my enthusiasm for the research despite all the adversity I experienced in the study.

I thank Dr. Dowd from Warwick Manufacturing Group, and Dr. Love from Aston university for providing invaluable comments, corrections and suggestions for improvement.

I should also like to thank the many individuals who have helped in the Ph.D programme, in particular, all the members of the simulation team.

Finally, particular thanks are due to my wife Fang Zhang for her enduring patience and support without which the work would not have been completed.

Contents

1	Introduction	1
1.1	Framework of production scheduling and control	2
1.1.1	Master Production Schedule (MPS)	3
1.1.2	Material Requirement Planning (MRP)	4
1.1.3	Shop Floor Control	5
1.1.4	JIT and OPT philosophy and frameworks of their produc- tion management systems	8
1.2	Maintaining shop floor stability	10
1.2.1	Problem and major approaches in maintaining short term production plan	11
1.2.2	Maintaining shopfloor stability by predictive and reactive scheduling	13
1.2.3	Problems in maintaining shop floor stability	14
1.3	Objective of the research	15
1.3.1	Dual resource constrained environment and event oriented control	15
1.3.2	Disruptive event oriented control	16
1.3.3	Major issues and objective of the research	17
1.4	Overview of the thesis	20
2	Review of approaches to shopfloor scheduling	21
2.1	Classical scheduling theory and its development	22

2.1.1	Classification of scheduling problems	22
2.1.2	Performance and objectives of scheduling	23
2.1.3	Assumptions and limitation of the classical scheduling theory	25
2.1.4	Approaches and techniques developed in classical schedul- ing theory	28
2.2	Review of approaches in shopfloor scheduling practice	29
2.2.1	Objectives of practical shopfloor scheduling systems	30
2.2.2	Shopfloor scheduling approaches in modern production man- agement systems	31
2.3	Recent trends to shopfloor scheduling	35
2.3.1	Review of approaches for predictive scheduling	35
2.3.2	Reactive scheduling	37
2.4	Predictive and reactive scheduling for promoting shopfloor stability	41
2.4.1	Schedule maintenance and schedule repair	42
2.4.2	Problems and approaches in maintaining shopfloor stability	43
3	Review of approaches to shop floor control	45
3.1	Objectives of shop floor control	45
3.1.1	Classification of objectives of shopfloor control	45
3.1.2	Maintaining shop floor stability	46
3.2	Major control functions in shop floor control systems	48
3.2.1	Functions for control of manufacturing environment	48
3.2.2	Major functions for control of progress of production . . .	51
3.2.3	Interaction and relationships between different type of con- trol functions	53
3.3	Selection of shop-floor control actions	55
3.3.1	Various control actions	56
3.3.2	Important factors in selection of control actions	57
3.4	Disturbances and resource flexibility	58

3.4.1	Disturbances in job shop and discrete batch manufacturing	58
3.4.2	Resource flexibility for dealing with disturbances	60
3.5	Approaches and systems for real-time control	62
3.5.1	Priority rule based dispatching	62
3.5.2	Knowledge based approach for Shop Floor Control	65
4	Control strategies for maintaining shop floor stability	69
4.1	Dispatching strategies	69
4.1.1	Problems with simple dispatching rules	69
4.1.2	Major factors or constraints for consideration in dispatching	71
4.2	Disruptive events	72
4.2.1	Major characteristics of disruptive events	72
4.2.2	Important issues for determining control strategies for deal- ing with machine breakdown	74
4.3	Discussion on approaches of multiperspective dispatching	75
4.3.1	Review of approaches related to multiperspective dispatching	75
4.3.2	Important issues on management of bottlenecks	77
4.3.3	Definition of bottlenecks	77
4.3.4	Critical characteristics of bottlenecks	78
4.3.5	Review of approaches for look-ahead	79
4.4	Conclusions on dispatching strategy	81
5	A proposed shop floor control strategy	85
5.1	Outline of the proposed shop floor control strategy	85
5.1.1	Paradigm of the proposed control strategy	85
5.2	Approaches for coordination of order related factors	88
5.2.1	Approaches for description of candidate jobs' due date pri- ority	88
5.2.2	Consequences of dispatching actions	90

5.2.3	Approach of 'look-road-ahead'	92
5.2.4	Function of the look-ahead mechanism	93
5.2.5	Approaches for coordination of the due date related factors	94
5.3	Approach of real-time capacity management	99
5.3.1	Definition of bottlenecks	99
5.3.2	Approaches for capacity management in dispatching	101
5.3.3	Overtime based capacity control	103
5.4	Approaches for determination of job's ultimate priorities	104
6	Structure and implementation of a simulated manufacturing environment to test the proposed control strategy	113
6.1	Purpose of the simulated manufacturing environment	113
6.2	Review of conventional approaches to discrete event simulation	115
6.2.1	Review of trends in development of conventional simulation approaches for manufacturing systems	116
6.3	Knowledge based simulation system	118
6.3.1	Review of important features of expert systems for shop floor control	119
6.3.2	Review of rule based approaches	121
6.3.3	Approaches for implementation of knowledge based simulation systems in shop floor control	125
6.3.4	Overview of important features of Knowledge Craft for implementation of the environment	127
6.4	Structure and implementation of the Simulated Environment	129
6.4.1	Structure and implementation of the simulation system in SME	131
6.4.2	Control module	140

7	Experiments	151
7.1	Objective and design of experiments	151
7.1.1	Important issues on design of experiments	151
7.1.2	The production system	153
7.1.3	The production plan	154
7.1.4	Control related elements	158
7.2	Results from the experiments	160
7.2.1	Results from the preliminary experiments	161
7.2.2	Results from detailed analyses	169
7.2.3	Discussion of results	180
8	Summary and conclusions	185
8.1	Summary	185
8.2	Conclusions	188
8.3	Further research	189

List of Tables

5.1	Information and control policies used in different situations	106
7.1	Mean times used in machine breakdown distributions	155
7.2	Results from experiments at 70% shop load level without overtime	162
7.3	Results from experiments at 77% shop load level without overtime	163
7.4	Results from experiments at 85% shop load level without overtime	164
7.5	Results from experiments at 70% shop load level with overtime . .	165
7.6	Results from experiments at 77% shop load level with overtime . .	166
7.7	Results from experiments at 85% shop load level with overtime . .	167
7.8	Mean values of results at 77% shop load level without overtime . .	171
7.9	Mean values at 85% shop load level without overtime	172
7.10	Mean values of results at 77% shop load level with overtime . . .	173
7.11	Mean values of results at 85% shop load level with overtime . . .	174
7.12	t-test (paired) at 77% shop load level	177
7.13	t-test (paired) at 77% shop load level for different breakdown levels	178
7.14	t-test (paired) at 85% shop load level	179
7.15	t-test (paired) at 85% shop load level for different breakdown levels	180
7.16	Mean and confidence interval for 77% shop load level (ODD - control strategy)	181
7.17	Mean and confidence interval for 85% shop load level (ODD - control strategy)	182

List of Figures

1.1	The major scheduling and control elements in a standard production management system	3
1.2	Shop floor control: An Integrated Framework	7
5.1	Jobs and workcentres for which information have been included in the proposed control strategy	87
5.2	The control process for identifying levels of effect of job selection .	96
5.3	Process of job selection in late job situation	107
5.4	Process of job selection in critical job situation	108
5.5	Process of job selection in due start job situation	110
5.6	Process of job selection in no adverse effect situation	111
6.1	The structure of the Simulated Manufacturing Environment . . .	130
6.2	Components of the Simulated Environment	133
6.3	Structure of physical element schemata	134
6.4	Structure of job and operation schemata	135
6.5	Structure of shift schema	135
6.6	Schemata used in control module	141
6.7	A rule for selecting late job by total tardiness	143
6.8	A rule for selecting critical job	144
6.9	A rule for selecting due start job by LRA control policy	145
6.10	A rule for selecting non due start job	146
6.11	Process of overtime scheduling	148

6.12	The main structure of hierarchical commands in user interface . .	150
------	---	-----

Abbreviation

AI Artificial Intelligence

CJ Critical Job

COVERT Cost Over Time

CRP Capacity Requirements Planning

CRR Critical Ratios

DM Downstream machine

DRC Dual Resource Constrained

DSJ Due Start Job

EDD Earliest Due Date

FCFS First Come First Serve

FMS Flexible Manufacturing System

GT Group Technology

IM Idle Machine

ISIS Intelligent Scheduling and Information System

JIT Just In Time

KBB Keeping Bottleneck Busy

LEX Lexicographic-sort

LRA Look Road Ahead

MEA Means-Ends-Analysis

MPS Master Production Schedule

MRP Material Requirement Planning

ODD Operation Due Date

OPIS Opportunistic Intelligent Scheduler

OPT Optimised Production Techniques

ORR Order Release/Review

RW Remaining Works

SME Simulated Manufacturing Environment

SPT Shortest Processing Time

TA Tardiness

TWK Total Work Content

WIP Work In Progress

Chapter 1

Introduction

As manufacturing companies are increasingly operating in a global competitive market, production management systems play a key role in helping them to meet ever increasing customer demand for better quality, more features and better delivery performance.

Production management systems deal with all of the activities from acquisition of raw materials to delivery of completed products, and their main functions are materials planning, resources planning, and synchronisation of resource and material plans to meet the objectives of production. The key activities are production scheduling and control.

Production scheduling can be defined as the allocation of available production resources over time to best satisfy some given set of criteria [Graves 1981]. Objective of production scheduling is to produce time-phased material requirement plans for production and purchases based on given production goals.

Production control is a function of management which directs and controls the material supply, puts into action the material requirement plans, monitors the execution of the plans, compares the results with the plans, reports on variances from the plans, and takes corrective actions to adjust the plans whenever neces-

sary.

Capacity planning is an important control activity in a production management system and the basic intent is to provide projections of the capacity needs implied by current material plan, so timely actions can be taken to balance the capacity needs with capacity available.

If there is a mismatch between available and required capacity, the management task is to change the capacity to meet the requirements, the requirements to meet the capacity, or some combination of the two. Capacity requirement can be changed by inventory changes, alternative routings, make or buy decisions, subcontracting, or changing customer promise dates. The manufacturing capacity can be increased by overtime, extra shifts, new equipments, etc. The choice of capacity changes is influenced by time horizon for the decision, cost, market position, flexibility, and institutional restrictions.

Production control aims to ensure that right material is available at the right time in the right place to the right specification, so that the products are manufactured in the right quantity, at an appropriate rate, in the most economic way.

The approaches and techniques for production scheduling and control are based on required accuracy, aggregation level, and ease of preparation.

1.1 Framework of production scheduling and control

In any company, production scheduling and control encompasses distinct phases

which are usually described as three levels: Master Production Schedule, Manufacturing Resource Planning and Shop Floor Control (Figure 1.1).

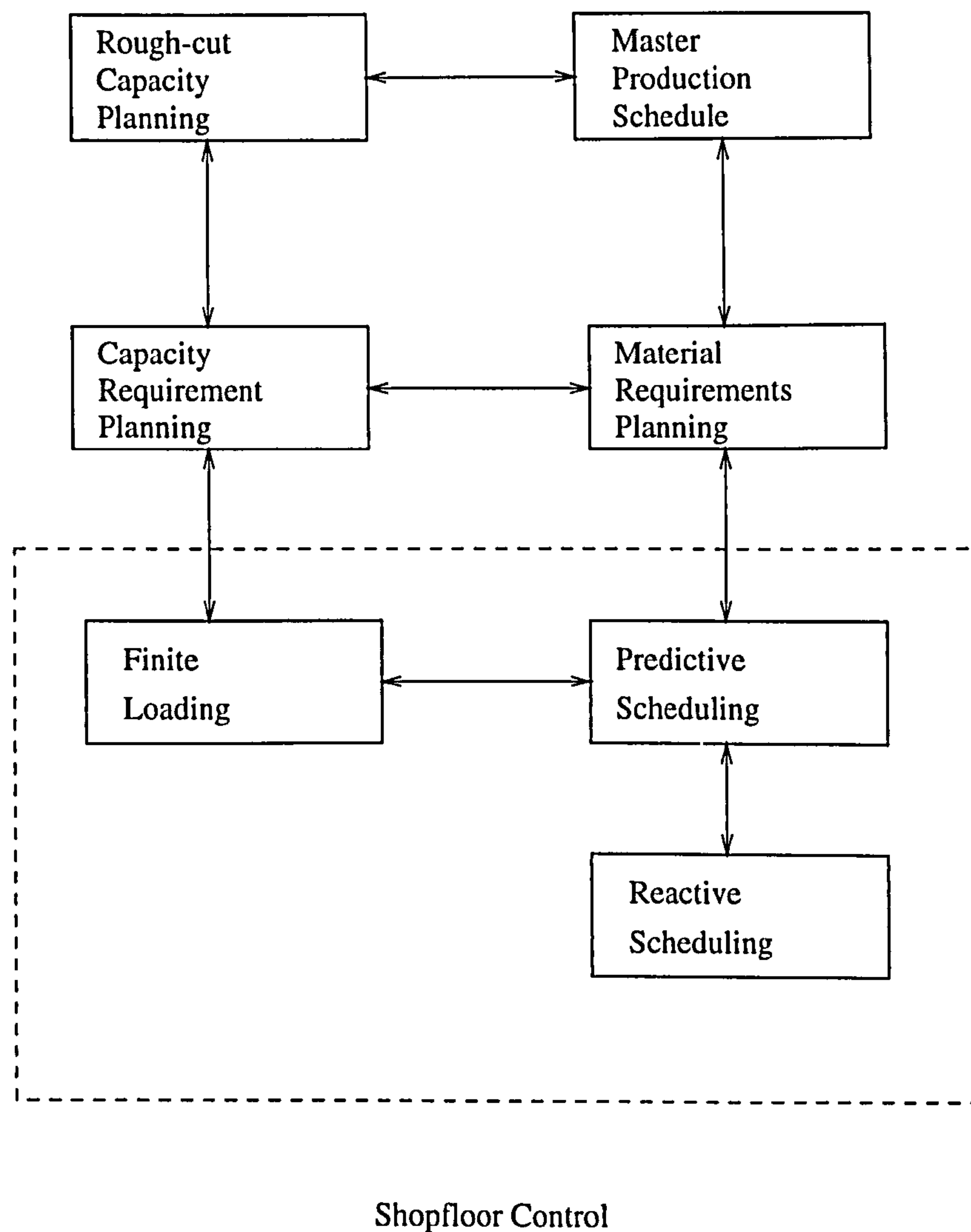


Figure 1.1: The major scheduling and control elements in a standard production management system

1.1.1 Master Production Schedule (MPS)

Master Production Schedule (MPS) is a statement of production which is defined

as a listing of end products to be manufactured, quantities of each end product and the dates they are to be delivered.

MPS provides a basis for coordination of market and production, and contains all significant demands on plant facilities. These demands include forecast demands and actual demands, orders from customers for products and service parts, warehouses etc. To coordinate manufacturing resources with MPS on an ongoing basis, and to identify future problems, such as bottlenecks, capacity limitation is taken into account by means of 'rough-cut capacity planning'.

Rough-cut capacity planning checks, at a macro-level, availability of capacity on critical resources, and evaluates potential capacity problems for a particular MPS to ensure the feasibility of the master production schedule. The information used pertain to availability of critical resources, production progress, restriction on the availability of critical materials, and inventory status.

The production requirements in a master production schedule could be sent to the shop floor for initiating manufacturing if the short term and medium term manufacturing resources can be guaranteed. But, in most practical production scheduling and control systems, more detailed material and capacity requirement plans are required before orders are released for production and/or purchase. And this is carried out by Material Requirement Planning.

1.1.2 Material Requirement Planning (MRP)

MRP is a computational technique for converting a master production schedule for end products into a detailed schedule for the raw materials and components used in the end products. MRP uses Bill Of Material to translate a period-by-period set of master production schedule requirements into subordinate com-

ponent requirements, uses fixed estimates of lead times, and takes into account inventory status for producing a resultant time-phased set of components/raw material requirements for manufacturing and purchase. MRP provides an order release plan for manufacturing and purchasing, i.e. when to place manufacture and/or purchase orders and for what quantities.

MRP can be used for long term and medium term material requirements planning, and feasibility of its material requirement plans can be checked by means of Capacity Requirements Planning (CRP).

CRP uses the material requirement plan from MRP (which includes all actual batch sizes, lead times for both open shop orders and planned order releases, current status of all work-in-progress and demands for service parts and other demands that may not be accounted for in MPS), to compute the capacity required to manufacture all the component parts. A CRP procedure would examine the status of all open shop orders, *estimate* how long they will take (setup, run and move) at particular work centres, and thereby derive when they will arrive at subsequent work centres. This would be repeated for all planned orders from the MRP data base.

Since a CRP procedure computes capacity requirements but does not guarantee feasibility of the plan, some work centres may be overloaded during production. Therefore, a more accurate projection of work centre capacity may be needed.

1.1.3 Shop Floor Control

Shopfloor control is at the operational level of production management systems, and “governs very short-term detailed planning, execution and monitoring of

activities needed to control the flow of an order from the moment that it is released until the order is filled and its disposition completed.” [Melnik et al 1985] The activities in shop floor control include order release, detailed assignment, data collection and monitoring, control and feedback, and order disposition. [Melnik et al 1985] outlines the relationships between these activities as in **Figure 1.2**.

As shown in Figure 1.1, finite loading, predictive scheduling and reactive scheduling are the key functions for capacity planning and control at this level.

Predictive scheduling is for generating a predictive schedule according to available capacity and orders received from a higher level of production management system. Predictive scheduling utilises routing and operations data to determine the manufacturing resources required to produce the quantities of a given product as specified in the planned order. This can then be used to anticipate potential conflicts for available capacity on the shop floor. The goal is to schedule activities such that only what is actually required is produced, when it is needed and in the correct quantity, and that the production schedule is feasible. Key scheduling activities include work centre loading, job sequencing and production scheduling.

Work centre loading relates scheduled jobs for each work centre to its anticipated capacity and identifies amount of work to be completed by each work centre; job sequencing is to introduce priority rating for orders to compete for manufacturing resources; production scheduling identifies the orders to be worked on, their sequence and their timing. It may specify product due date and/or more detailed individual operations’ start and completion time.

Finite loading can be used for capacity management in predictive scheduling. Finite loading is a forward scheduling procedure which simulates actual job order starting and completion to produce a detailed schedule (work-to-list) for each shop order and each work center based on its finite capacity limit according to

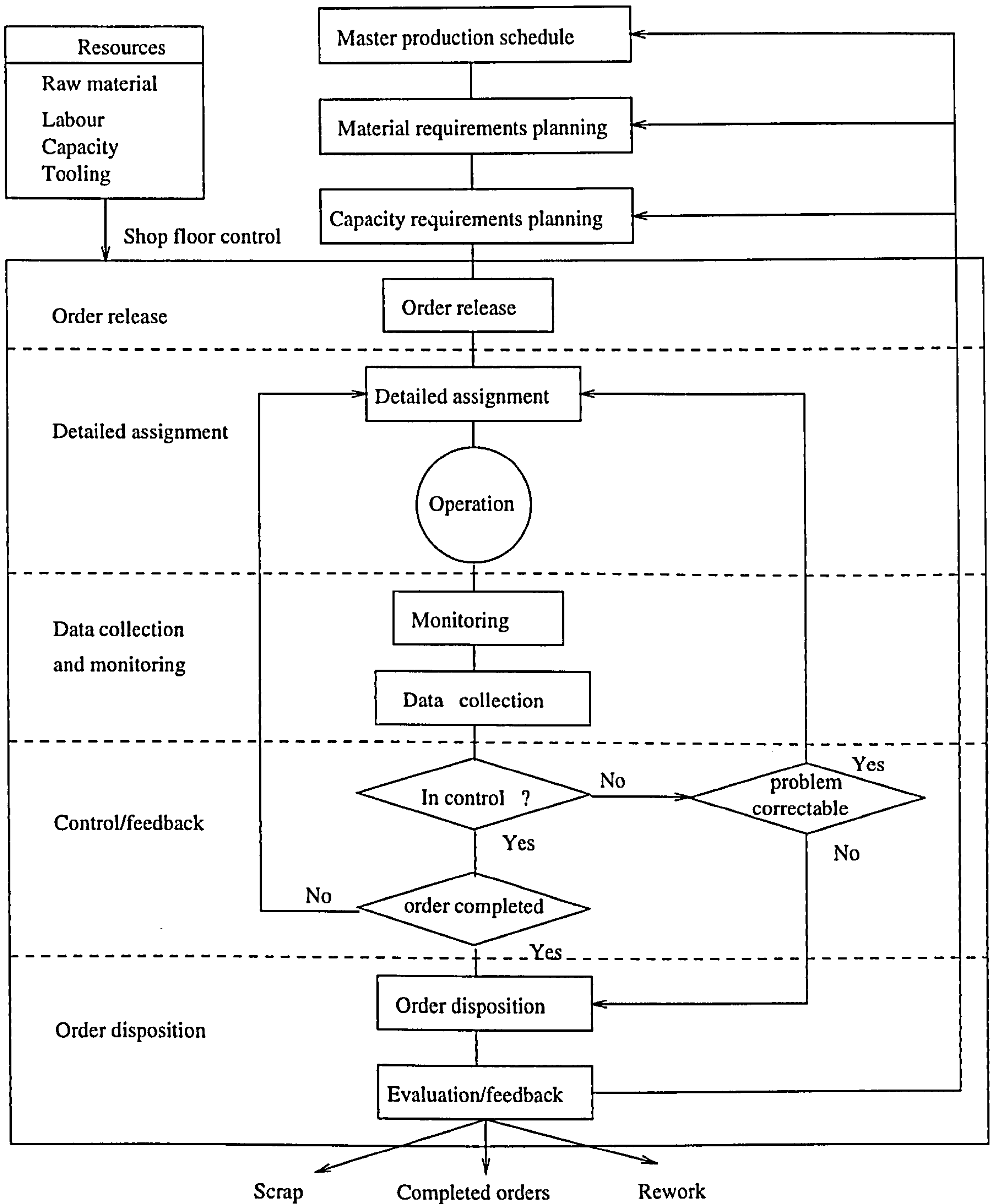


Figure 1.2: Shop floor control: An Integrated Framework

some priority rules. The result is a set of start and finish dates for each operation at each work center.

Finite loading and predictive scheduling are the major functions in order release, and in many cases, predictive schedulers are essentially finite scheduling mechanisms.

Reactive scheduling is the control activities to carry out resource allocation and to cope with uncertainty and random disturbance on shop floor. Detailed assignment and control (capacity control), as shown in figure 1.2, are the major mechanisms to carry out reactive scheduling. The detailed assignment is responsible for allocation of resources to the orders' operations, and capacity control is for short-term adjustment of resource capacities available on the shop floor.

One of the major tasks for reactive scheduling is to react to disruption events as deviations occur from the predictive plan. Its objective is to bring the production progress back to working plan based on pre-determined goals.

As shown in figure 1.2, shop floor control includes other activities: data collection and monitoring for collection of shop-floor information and monitoring overall progress of orders against the plan; feedback for sending information pertaining to the actual progress of orders on the shop floor to the planning system; and order disposition to dispose the orders after they are completed or scrapped.

1.1.4 JIT and OPT philosophy and frameworks of their production management systems

In a standard production management system, as shown in figure 1.1, the distin-

guishing features at each level are its function, time horizon employed, the detail of data used, and the different degrees of planning and control.

There is a vertical as well as horizontal relationship among the modules in an integrated, standard production management system described above. The vertical (i.e. level) relationship describes time range of material and resource planning which is from long-range to day-to-day, and detail of the material and resource planning from rough and gross to net and detailed. Horizontal relationship among the modules at each level is that between production planning and control activities.

The functions of the modules in a production management system can be changed or combined for specific production environments and/or market demands. JIT (just-in-time) and OPT (Optimised production techniques) [Goldratt & Cox 1984] are modern production management techniques having such a modified framework.

JIT is a philosophy covering all aspects of a manufacturing operation with the aim of producing only what is required and at the time it is required, with perfect quality and no waste. These objectives are usually described as follows: finished goods are manufactured just-in-time to be delivered; sub-assemblies are manufactured just-in-time to be assembled into finished goods; parts are fabricated just-in-time to go into sub-assemblies; materials are purchased just-in-time to be transferred into finished products.

Based on the assumption that demands are relatively stable and there is no serious shortage in capacity, JIT eliminates large portions of standard production management systems, including Rough cut capacity planning, MRP, predictive scheduling, etc. JIT also simplifies shopfloor control mechanism by exposing and eliminating quality and disturbance problems. Reactive scheduling in JIT is mainly restricted to responding to demands in such a way that each manufactur-

ing stage “pulls” the supply of items and resources according to its own needs from stages upstream.

OPT is a centralised shopfloor scheduling system which can directly use information from MPS, and data from Bill of Material and routings to generate schedules for production. OPT combines finite loading with infinite loading function, which are respectively responsible for producing bottlenecks resources schedules and non-bottlenecks resources schedules. Objectives of OPT is to maximise material flow in shopfloor by identifying bottleneck work centers and optimising utilisation of bottleneck resources according to specified goals.

To achieve the predictive outcome, OPT (like some other predictive scheduling tools) addresses robustness of schedule through the use of physical buffer and ‘time buffer’.

Importance of maintaining a stable production plan has long been realised by manufacturing industry.

1.2 Maintaining shop floor stability

In repetitive manufacturing environments with stable demand, long term production plans can usually be met without much concern about short term fluctuations of material supply and resource utilisation, as is the case with an effective JIT based production management system. But, in many manufacturing systems, e.g. job shop and discrete batch productions, detailed short term plans are required and generated at discrete time intervals.

Although changes in a production plan are inevitable due to changes in the pro-

duction environment or in response to customer requirements, many companies try to reduce these changes by setting time fences and/or freezing a short term plan since “To achieve the level of productivity necessary to remain competitive, stability in short-range manufacturing plans is essential” [Vollman et al 1990].

1.2.1 Problem and major approaches in maintaining short term production plan

In its current state of application, predictive scheduling is carried out at discrete intervals (e.g. week, day), to produce a plan which is achievable, good or even “best” if circumstances during the production process do not change.

The major difficulty inherent in shop floor control is that production operates in a dynamic environment which is often subject to high levels of uncertainty and unplanned events, and the predictive plan is subject to change. These events may be new available information from the higher level production management system in response to changed demands (e.g. new orders), and/or random events on the shop floor, such as machine breakdowns. Effects of these events could be propagated and magnified in the production system through dependent events and resulting interactions between resources and products.

In general, there are three ways to cope with these unexpected events in shop floor control:

- expose and eliminate sources of the events,
- robust predictive schedules,
- reactive scheduling.

The extent to which the first policy (expose and eliminate sources of the random

disturbances) can be relied upon, is very much dependent upon the manufacturing environment and the production management system. In a repetitive manufacturing environment with an effective JIT system, such a policy can work well since sufficient capacity is usually made available and most sources of uncertainty are at a reasonable level.

In OPT and other predictive planning tools, robustness of the predictive schedules is addressed through the use of appropriate levels of 'time buffer'. These capacity allowances could smooth small load fluctuations (e.g. variation of operation times), and reduce imminent impact of some disruptive events, but since it is impossible to predict occurrences of these events (exact types of events, time of occurrence, place of occurrence, etc), effects of these events can still be propagated and/or accumulated in the manufacturing system. Therefore, it often becomes necessary for reactive scheduling to modify the existing schedule or generate a new schedule during production. In other words, a complete solution to the scheduling problem in manufacturing systems, must include the function of predictive planning coupled with an ability to alter schedules in response to changing conditions on the shop floor.

Reactive scheduling can be used in different ways in shop floor control:

- shop floor scheduling can be totally reactive throughout the production process, if there is no predictive scheduler in the shop floor control, e.g. in JIT based production management systems,
- shop floor scheduling becomes totally reactive, after a predictive schedule is discarded as a result of an occurrence of a disruptive event (or some events);
- reactive scheduling works with a predictive schedule.

The last method was suggested by [Roy 1993] for maintaining shop floor stability.

1.2.2 Maintaining shopfloor stability by predictive and reactive scheduling

Maintaining shopfloor stability is an objective of shop floor control. As described by [Roy 1993], it aims to maintain the original plan as far as feasible.

Predictive scheduling is for generating a predictive schedule which is “realistic, efficient and robust and provides sufficiently clear information against which control can be exercised” [Roy 1993]. A detailed timing plan in the schedule will allow the control system to closely monitor production progress and conduct effective control; ‘time buffers’, will form ‘capacity cushions’ which could smooth small fluctuations on work loads as well as reduce the impact of disruption events.

The objective of reactive scheduling is “to bring the status of the manufacturing system, as far as feasible, back to its original plan” [Roy 1993].

The role of the reactive scheduler was described by [Bhattachayya et al 1991] as follows:

- i) assess the extent of deviation that may have occurred from the original plan,
- ii) devise an action plan to get the manufacturing system back to its predictive schedule, if possible;
- iii) modify the plan, if recovery to original plan is not feasible.

Capacity control, e.g. overtime, is also needed as part of a reactive scheduler for compensation of capacity losses caused by disruptive events.

A key issue for maintaining shop floor stability is to smooth the flow of work by identifying and managing capacity problems caused by disruptive events.

1.2.3 Problems in maintaining shop floor stability

In practice, detailed assignment is usually carried out by a dispatching mechanism, and dispatching strategy is formally described by a priority rule, e.g. based on operation due date (ODD). Dispatching is seen as a major real-time shop floor control function by many researchers, but in industrial applications, the priority rule based dispatching system is unable to detect capacity problems, e.g. congestion and ‘moving’ bottlenecks, caused by disruptive events. Effects of these events could be propagated and accumulated in the system, and result in peak and valley work loads. Some machines may be starved of work, and some may have long queues. These capacity problems will sooner or later prevent orders accessing required resources in a timely manner, and some orders may miss their due dates.

It is the shopfloor expeditor, supervisor or foreman’s job to cope with contingency in real time. But their actions often take place when some orders have obviously lagged behind their due dates, e.g. expediting critical orders. These ‘spur-of-the-moment’ revisions to the schedule sometimes result in continual rescheduling and instability on the shop floor. For example, as observed by [Melnik et al 1985], “in most instances the use of expediting does not significantly improve the overall performance of the shop floor.”

What we need is a reactive scheduler which can not only take into account order related factors in decision making, but is also able to detect resource related problems and smooth fluctuation of work loads, i.e. the scheduling mechanism should not only schedule job based on order but also resource related information. The importance of including and appropriately coordinating these perspectives have been addressed by some researchers.

OPT is a typical example of co-ordinating these two perspectives in generating schedules. OPIS (Opportunistic Intelligent Scheduler) [Smith & Ow 1990] is an integrated framework for generating and revising factory schedules by multi-perspective scheduling. [Melnik 1988] suggested such modification to dispatching procedures using information on both orders and resources, and the approach “pushes research into global-based dispatching rules - a direction generally scorned in past research studies.”

In this research, a control strategy for dispatching orders based on both order perspective and resource perspective is investigated.

1.3 Objective of the research

The research is directed at the study of control strategy in a dual resource manufacturing environment.

1.3.1 Dual resource constrained environment and event oriented control

Dual-resource constrained (DRC) systems represent typical job shop and discrete batch manufacturing systems. In DRC systems, there are two forms of capacity constraints: machine and labour, and in most manufacturing environments, the number of workers is less than the number of machines. “Studies on problems of dual-resource constrained (DRC) systems is one of major attempts to expand the focus of shop floor control research with capacity involved” [Trelevent 1989].

A DRC system is much more difficult to model than the machine limited system because there is a need for both a labour assignment strategy for assigning

workers to workcentres, i.e. queue selection, and a job selection policy on a given workcentre. The dispatching mechanism in a DRC system needs to include both strategies.

In DRC systems, effects of disruptive events can be propagated through both machine and labour, and the extent of deviation of the production process from a working plan is more difficult to detect in real-time shop floor control.

Both machine flexibility and labour flexibility is important for dealing with disruptive events. A dispatching mechanism in a DRC system should have and use such information in the control strategy.

1.3.2 Disruptive event oriented control

Shop floor control strategies can be classified by the way they react to the changes in the manufacturing system.

In some control strategies, e.g. expediting and priority rule based dispatching, various disruptive events are not anticipated and explicitly represented, and the control action is based on the detected local changes. Since lack of information about disruptive events, these control strategies, though fast in execution, are shortsighted and not effective in many situations.

Shop floor control can also be disruptive event oriented, i.e. information about disruptive events is used in its decision making. In industrial environments, shop floor personnel, (operators, dispatchers, supervisors, etc) have the knowledge and strategies to cope with different disruptive events and/or different situations on shop floor, and they frequently use this knowledge in their practice of shop floor control to improve system performance. For example, when a workcentre is broken down, its downstream machine will be given priority to assign labour for preventing congestion after the broken down workcentre has recovered. For

effective shop floor control, such information and knowledge is frequently used in dispatching process in industrial environment, as observed by [Melnik 1988].

Machine breakdown is one of the most commonly occurring disruptive events on the shop floor. Machine breakdown is a difficult problem to deal with due to its randomly occurring nature as well as the ways by which it would affect system performance. A machine breakdown can affect production process during the time in which the machine is broken down, e.g. delay the job on the broken down machine, and after the machine has recovered, such as congestion at the workcentre. Machine breakdowns can also have a global impact on the production process. A machine breakdown can have a direct effect on jobs on the machine, making its workcentre lose available capacity, and indirectly affect other jobs and workcentres in the system, such as both upstream and downstream machines. There is no satisfactory solution reported in the literature, which can be generally used for control of job shop or discrete batch production subject to machine breakdowns.

This research, therefore, tried to study shop floor control strategies for maintaining shop floor stability in manufacturing environments subject to machine breakdown.

1.3.3 Major issues and objective of the research

For maintaining shop floor stability in manufacturing environments under the impact of machine breakdowns, the shop floor control systems need to constantly events, and then select corrective actions. Since effects of a disruptive event can be propagated and magnified through interaction between resources and cascaded conflicts between orders for timely access to required resources, a major task of a shop floor control system is to prevent or reduce such effects, and re-establish, if possible, synchronised work flow.

Dispatching has long been seen as a major control activity on the shop floor by researchers. But since most of this research has focused on studying highly simplified and a small set of production problems which unfortunately have little in common with real world factory environments, in many discrete manufacturing environments, real-time shop floor control, as described in section 1.2.2, is still based on simple priority rules which do not use real-time information plus a ‘firefighting’ style of reactive scheduling.

It has been realised that dispatching should be based on real-time information. As pointed out by [Browne et al 1988], “The dispatcher is, in one sense, a real-time scheduler which assigns jobs to work centres based on real time information, the present status of the shop floor and on the priorities by the scheduler.” [Melnik 1988] also observed that in effective shop floor control systems, “priorities generated by the dispatching rule were only inputs to the dispatching process”, and the dispatcher should be given some flexibility in job selection based on real-time information, especially information on upstream and downstream work-centres. [Melnik 1988] noticed that “The impact of the dispatcher was greatest when was given visibility over upstream and downstream work loads.”

This research aims to propose a modified dispatching procedure which would include real-time information (both order related and resource related) in decision making for promoting shop floor stability. As system performance can be affected by many factors, a major issue in the research is to identify, describe (quantify if necessary), and find a way to coordinate the factors which could significantly affect system performance. These factors would include organisation goal related information (maintaining shop floor stability in this research), e.g. tardiness and number of tardy jobs; and physical constraints, e.g. machine breakdown; resource availability; etc. A control strategy is required for describing these factors and addressing a way by which these factors could be appropriately used in the dispatching process.

Since such control knowledge has a high heuristic content, the control process in such a dispatcher should be a heuristic procedure. Therefore, a major objective of this research is to develop a heuristic dispatching procedure, which has the ability to take consideration of both order and resource information for effective control. Such a dispatcher would not only have a broader view on shop floor status, but also include knowledge to cope with disruptive events (in this study, machine breakdown related knowledge). The multiperspective dispatcher must also have the potential to be extended for manufacturing environments with other random events, such as rush jobs and labour absence, in addition to machine breakdowns.

For testing and demonstrating performance of control strategies, a special simulation tool with the ability of efficiently modelling and manipulating heuristic knowledge is required. As there is no such tool available for this study, another major task in the study is to develop such a simulation tool.

Such a tool needs a scheduler for generating detailed, short term production plans (predictive schedules); an emulator for modelling physical structure and material movement and operation of the equipments in the system; a control system for modelling shop floor control strategy (including dispatching, capacity control, etc); a monitor/data collector for monitoring the shop status and translating and passing this information to the control system and collecting data for statistics; and a user interface for modelling and manipulation of system models, and conducting experiments and outputting results from experiments.

In order to test the effectiveness of the proposed control strategy, some hypothetical manufacturing system models are required for capturing the essential characteristics of job shops in general. The experiments need to be carried out in a variety of system settings and conditions, e.g. different level of machine breakdowns and different levels of shop utilisation. Another major work in this research is then to study experiment related issues, e.g. design of experiments and evaluation of experimental results.

1.4 Overview of the thesis

This thesis is organised as follows: chapter 2 reviews production scheduling theory and recent research in industrial scheduling practice, especially in maintaining shopfloor stability. Chapter 3 reviews approaches and techniques which have been used for shop floor control. Chapter 4 reviews approaches and techniques of shop floor control which closely relate to the proposed control strategy, and outlines the control strategy. Chapter 5 describes in detail the proposed control strategy. Chapter 6 describes the simulated manufacturing environment that has been developed to test the strategy, its structure, approaches and techniques. Chapter 7 includes description of the experiments, objective and design of the experiments, presentation of the results, and analysis and discussion. Chapter 8 summarises conclusions from the study and includes suggestions for further research.

Chapter 2

Review of approaches to shopfloor scheduling

Scheduling problems typically involve a set of jobs to be processed, where each job has a set of operations to be performed. Operations require resources such as material, machines, labour and tools, and must be completed according to some feasible technological sequence defined for the jobs. Developing a production schedule involves designating production batch sizes and assigning the times at which each operation in the routing will receive required resources for it to start.

The shopfloor scheduling problem has been studied by theoreticians in operations research and management, and industrial practitioners for the past several decades for two rather different purposes, theoretical development and industrial application. The industrial practitioners view shopfloor scheduling as part of a production management system, and aim to improve the quality, consistency, and acceptability of production schedules through improving the production management system, its structure, communication technology and so on, while the mathematicians in operations management attempt to construct formal mathematical models and devise methods for finding optimal solutions for these models for theoretical achievement.

In classical scheduling theory, scheduling problems are usually simplified as machine-scheduling or job-shop problems for capturing the fundamental computational complexity of the central problem of sequencing jobs on machines.

2.1 Classical scheduling theory and its development

The job-shop or machine-scheduling problem is usually stated as follows: N jobs have to be processed on M machines. Each job consists of a set of operations which are to be processed by a subset of M machines in a unique order. The scheduling problem is to determine the sequence and timing of each operation on each machine, such that some given performance measure can be optimised.

2.1.1 Classification of scheduling problems

An useful classification was given by [Graves 1981], in which scheduling problems have been classified by five dimensions:

Requirements generation - Scheduling problems can be categorized by identifying the source of the requirements. In an open shop, requirements are directly generated by customer orders, while in a closed shop, requirements are indirectly generated by requests from inventory.

Processing complexity - which is concerned primarily with the number and routing restriction of operations for each job. The number of operations is either single, i.e. one-stage, or multiple, multi-stage. The flow patterns of jobs are either identical, i.e. operations have to be processed on the same set of machines with the same sequence, or non-identical.

Scheduling criteria - indicates the measures, which are related to either performance or cost, by which the schedules can be evaluated.

Requirement specification - indicates how the requirements are generated in the scheduling problem. Either all the requirements are known, i.e. deterministic, or there are random variables in specification of the requirements, i.e. stochastic.

Scheduling environment - The scheduling environment can be classified as either static or dynamic by the assumptions on the availability of information on future requirements.

[Conway et al 1967] suggested a four-parameter notation, written as A/B/C/D, which is often used for identification of individual scheduling problems in classical scheduling studies with the following meanings:

A: describes job arrival process, e.g. a distribution for the description of the time interval between job arrival, or the number of jobs which arrive at the shop simultaneously at the beginning,

B: number of machines in the shop,

C: flow pattern in the shop, e.g. job shop by J, flow shop by F, etc.

D: the criterion by which the schedule is to be evaluated or to be optimised, such as T for mean tardiness, F for mean flow time, etc.

In this description, the criterion described by D is seen as a performance measure and objective of scheduling.

2.1.2 Performance and objectives of scheduling

[Mellor 1966] lists 27 different objectives, which include three types of decision-making goals that are commonly used for measurement of schedule performance: efficiency, flexibility and meeting deadlines.

Efficiency means efficient utilisation of resources and can be measured by resource utilisations, and the major criteria in this category cited by Mellor are:

- "Minimum idle facility investment.
- Minimum facility set-up costs.
- Maximum weighted facility utilisation.
- Maximum utilisation of manpower.
- Optimal assignment of various labour grades."

Flexibility is often viewed as responsiveness built into a schedule to respond to changes in demand. Some of the criteria in this class cited by Mellor are:

- Sensitivity to possible production changes.
- Reserve capacity for rush orders.
- Ability to permit arbitrary job priorities, such as in dealing with preferred customers, emergency parts, etc.

The major criteria for meeting deadline cited by Mellor are:

- Adherence to promised shipping date.
- Achieving production target within specified time.

In scheduling theory, as noted by [Baker 1984], the first type of goal, efficient utilisation of resources, is often described by maximum completion time (or makespan), the second described by rapid response to demands, and measured by mean completion time, mean flow time, or mean waiting time, the third by mean tardiness, maximum tardiness and the number of tardy jobs.

As pointed out by [Graves 1981], classical scheduling theory usually addresses single criterion problems, which are either schedule performance, or cost. In the literature, reported studies for open shop problems deal primarily with schedule performance criteria, whereas the studies of closed shops are concerned with a minimum cost criterion. In real production environments, however, schedule evaluation is often based on a mixture of both cost and performance criteria.

For design of constructive solutions and optimisation algorithms in the general case, many assumptions are often made in classical scheduling studies.

2.1.3 Assumptions and limitation of the classical scheduling theory

As summarised by [McCarthy & Liu 1993], following assumptions appear frequently in scheduling theory literature to simplify scheduling problems:

- (1) “Machines are always available and never break down.
- (2) Each machine can process at most one job at any time.
- (3) Any job can be processed on at most one machine at any time.
- (4) Ready times of all jobs are zero, i.e. all jobs are available at the commencement of processing.
- (5) No pre-emption is allowed, i.e. once an operation is started it is continued until complete.
- (6) Setup times are independent of the schedules and are included in processing times.
- (7) Processing times and technological constraints are deterministic and known in advance and so are due dates, where appropriate.”

Though some of the assumptions reflect the reality of industrial production systems, (e.g. in many manufacturing systems, each machine can process at most one job at any time), others, such as machines are always available and never break down, are not. It is well known that machine breakdown is one of the most commonly encountered disruptive events on the shop floor. As shown in statistics collected by [Sharma 1987], the average down time per machine per month could be over 40 hours on some occasions.

As pointed out by [Graves 1981], for most production environments the scheduling problem is stochastic and dynamic; most models for scheduling problems, however, are inherently deterministic and static. In other words, machine scheduling itself is too restrictive a formulation to provide results for actual production scheduling. Some of these major restrictions are listed as follows:

- In machine scheduling theory, only one type of resource, machine, is considered; in real production systems, however, other type of resource(s) are usually required for job processing, such as workers and/or tools.
- In machine scheduling theory, the dynamic nature of manufacturing systems is usually described by random arrival of jobs, and the scheduling environment is assumed to be static, i.e. there are no random disturbances. In industrial environments, however, unexpected events constantly disrupt a shop, which is rarely stable for more than a short period [McKey et al 1988].
- In machine scheduling theory, a single optimality criterion, which is either schedule performance, or cost, is usually used to describe the goal or objectives. But, in most production environments, schedule evaluation is based on a mixture of both cost and performance criteria, such as meeting customer due dates with minimum production cost, and practical scheduling systems must manage to capture and balance a great variety of goals at the same time.

- In classical scheduling theory, as noticed by [McCarthy & Liu 1993], the planning time horizon in which the scheduling problem is being studied is generally not considered. Though it is assumed to be short term, there is no consideration of shift pattern in the scheduling theory, there is no flexibility for capacity adjustment, e.g. overtime. In most real manufacturing environment, however, shift system must be taken into account when planning and scheduling resources, and there will be some flexibility to adjust resource capacity when it is necessary.

Since classical scheduling theory has tended to consider scheduling problems in isolation, models studied are usually deterministic and static which has not taken into account the interaction among various functions in production management systems, such as higher levels of decision making and random disturbances on the shopfloor. As a result, real pressures on the scheduler, such as the dynamic nature of many environments, capacity planning and load balancing, etc, are often ignored in scheduling theory.

In a real world manufacturing environment, however, many different elements, as discussed in chapter 1, ranging from organisational structure of its production management system to types of production processes, may influence the nature of the scheduling function: the objectives, criteria and constraints within the function and the reality of scheduling practice.

Though in general, classical scheduling theory has had little impact on the real world factory scheduling problems to date, some methods and techniques which have been developed or tested in scheduling theory study can be used for solving scheduling problems in industrial environments.

2.1.4 Approaches and techniques developed in classical scheduling theory

[Conway et al 1967] classified the approaches in scheduling theory as Algebraic, Probabilistic and Monte Carlo. The first two approaches are categorised as optimal solutions, the last one as non-optimal. A variety of techniques have been used in scheduling theory study, such as enumeration, queuing network, critical path analysis, priority rules and simulation.

The theoretical work for exploitation of the special structure of the machine scheduling problem has provided a great understanding of the nature of these problems. [Lenstra et al 1977] found that most machine scheduling problems are NP-hard i.e. the time required to compute an optimal schedule increases exponentially with the size of the problem when there are more than two jobs and two machines, and fast optimal algorithms are unlikely to exist for these problems.

With the realisation of the limitation of optimisation approaches, classical scheduling theory showed a tendency to emphasise the practical nature of scheduling problems and to try to bridge the gap between the theory and practice since 1980s [McCarthy & Liu 1993]. [McCarthy & Liu 1993] classified optimal approaches as enumerative optimal, and efficient optimal which generates an optimal schedule in polynomial time with respect to some scheduling criterion.

One of the important contributions in scheduling theory study is the development of some heuristic approaches. A large number of single-stage heuristics, ranging from simple priority rules to more complicated heuristics, have been developed and tested for dispatching jobs, i.e. “selection and sequencing of jobs to be run at individual work centres and the authorisation or assignment of work to be done.” [Melnik et al 1985]

The solution strategies and approximation algorithms derived from exploiting the heuristics provide useful information which forms the basis of some practical scheduling applications and research. For example, many finite scheduling tools, e.g. PROVISA [Marriott 1994], have included the dispatching rules tested in scheduling theory study, e.g. EDD (Earliest Due Date). The look-ahead heuristics suggested by [Gere 1966], have also been used by some researchers in their studies, [Zeestraten 1990], [Ben-Arieh et al 1989].

Simulation approaches have played an increasingly important role in scheduling theory studies, especially as an evaluation tool for comparison of these different heuristic methods and rules.

Information on heuristic approaches in scheduling theory studies has also offered a readily accessible starting point for research into the development of algorithms using the new methodology. For example, as [White 1990] noticed, much interest in AI (Artificial Intelligence) approaches to production scheduling is closely associated with the development of novel heuristics for combinatorial sequencing problems.

2.2 Review of approaches in shopfloor scheduling practice

In contrast to the pursuit of mathematical models and optimal solutions, practical scheduling approaches concentrate on solving production scheduling problems encountered in real industrial environments, and aim to seek feasible and satisfactory or good solutions for different manufacturing settings.

In industrial environments, a shopfloor scheduling system is part of a production management system, and it must follow the goals which have been set by

its manufacturing organisation. These goals describe the overall concern of the organisation: desire to make scheduling decisions that satisfy customers and maximise profits.

2.2.1 Objectives of practical shopfloor scheduling systems

[Mellor 1966] lists 27 different objectives, which included three types of goals that are commonly used for measurement of schedule performance: efficiency, i.e. efficient utilisation of resources; flexibility which is viewed as responsiveness built into a schedule to respond to changes; and meeting deadlines.

In current scheduling practice, they are usually described by maximising resource utilisation, minimising WIP (Work In Progress), and meeting due dates.

Manufacturing industries wish to achieve all the goals at the same time, but these goals are often conflicting, e.g. the objective of meeting due dates could conflict with maximising utilisation of resource. In practice, some of these goals are more preferred in particular manufacturing organisations, and this is usually determined by market competition requirements, manufacturing philosophy adopted and the current available technology.

For a long period of time, manufacturing industries concentrated on improvement of efficiency through automation of equipment and economies of production scale. The major challenge to modern manufacturing, however, comes from the rapid changes in customer requirements and demands. The importance of meeting customer requirements and flexibility have greatly increased. Modern production management systems, therefore, need to respond rapidly to customer requirements, and the shopfloor scheduling systems need to be more flexible to accommodate the changes.

Meeting customer due dates is a major concern of a company in meeting customer

requirements. Order due date is an important part of the agreement between the customer and the company as orders are received. A late order affects customer satisfaction and the likelihood of future business. Flexibility, however, is also important, since this feature would provide a guarantee or, at least, increase the possibility of meeting customer requirements in dynamic environments.

In many shopfloor scheduling systems, (typical of which are MRP/MRP II based production management systems), predictive schedules are usually created at fixed intervals. Each schedule is generated for a short period of time, e.g. a week. After this period, another predictive schedule is needed to control production for the next period, and so on. For consistency in the production process and predictability of behaviours of the supply chain, it is important for manufacturing systems to adhere to a predictive plan and to complete preset tasks by the end of the period, i.e. 'hit the schedule', as well as to accommodate the changes to the schedule due to new information from higher level of planning and control, such as rush jobs and/or random events on the shopfloor like machine break down. From this view, maintaining shopfloor stability was described by [Roy 1993] as a goal to emphasise the need to maintain a predictive production plan. [Fox & Smith 1984] described this goal as "minimising the amount of disruptions to shop operations caused by revisions to the schedule" from rescheduling point of view. [Roy 1993], however, described this goal from another dimension, that of schedule adherence, he pointed out "a degree of shopfloor stability, consistency and predictability is required for effective management, and shopfloor control should be concerned with schedule adherence, with its objectives set by the predictive plan itself".

2.2.2 Shopfloor scheduling approaches in modern production management systems

Since the shopfloor scheduling function is part of a production management system, the complexity and structure of practical shopfloor scheduling systems are to a large extent determined by the manufacturing philosophy adopted, and tech-

niques used in its production management system. As briefly stated in chapter one, the most current of the manufacturing philosophies and associated software packages are MRP and its successor MRP II, and OPT.

MRP and MRP II

MRP systems are perhaps the most widely installed Computer Aided Production Management (CAPM) software in industry today.

An essential and core function of MRP is to explode demand from an end product down to its components, and with computer power, MRP can perform both detailed bookkeeping and extrapolation functions. MRP is needed to provide the expected future resource requirements and to make decisions on lot sizes according to current information on the present shop status.

The MRP/MRP II paradigm showed that hierarchical planning, with multiple levels of the manufacturing process, is a highly effective means for coping with the complexity and variety of manufacturing systems. Through such a hierarchical structure with a centralised computer and manufacturing database, the work of people in many different manufacturing functions can be better coordinated, and a large volume of common information can be shared. In this way, MRP II not only can significantly facilitate the manufacturing planning task, but also plays an integrating role in a manufacturing organisation.

From the shop floor scheduling point of view, however, MRP/MRP II is, in essence, a backward, infinite scheduling tool, which could prescribe machine loading in excess of 100% and production volumes, capacity or due dates may need to be adjusted at shopfloor level in order to produce feasible schedules.

MRP II is an off-line tool and entirely deterministic. Since it is difficult for MRP/MRP II to anticipate the impact on schedules of dynamic manufacturing

environments, such as random events, MRP is sometimes referred to as a type of push system in production control, by which production schedules are devised off-line and are used to push jobs through the facility, irrespective of current information on production status.

While fast advancements in computer technology may overcome the latter problem in the future, (e.g. [Fisher 1996] reported having reduced the cycle from 36 days to three days by using a ‘fast’ planning system connected to MRP), to overcome the former problem needs some fundamental changes on logistics of the MRP system because MRP systems “do not inherently make scheduling decisions; they have no mechanisms for considering the standard tradeoff associated with scheduling decisions” [Graves 1981].

As briefed in chapter 1, shop floor scheduling in MRP/MRP II based production management systems is carried out at the shop floor control level.

OPT

To attain the goal for making money, OPT considers activities on the shopfloor to be critical. Therefore, shopfloor issues, such as bottlenecks, set-ups, lot sizes, priorities, random fluctuations and performance measurements, are treated in great depth. Basically, OPT can be considered from two points of view - the OPT philosophy for manufacturing planning and control and the OPT software product.

The OPT philosophy was described in terms of ten relatively simple rules by [Goldratt & Cox 1984], and its objective was described as to reduce inventory and other operating costs while simultaneously increasing the throughput of the manufacturing plant. These OPT rules present very useful insights into the cost implications of scheduling decisions on the shop floor, and can be usefully applied in shopfloor control systems without the support of the OPT software product,

e.g. the importance of recognising bottlenecks, and discriminating between bottlenecks and non-bottlenecks in an attempt to manage the operations of the shopfloor.

The function of the OPT software is to automatically generate an 'optimum' schedule, typically for job shop and discrete batch production by its unique scheduling algorithm. Critical resources or bottlenecks are identified first. Then the operations that use the bottleneck resources and critical resources, and all operations that follow these resources, are scheduled first by a secret algorithm; and all the other operations are scheduled backwards thereafter.

OPT represents a tendency to emphasise a technical solution to some very complex organisational as well as technical problems in production management. Application of the OPT software in manufacturing industries, however, has so far been somewhat limited. One of the reasons is that the method of schedule generation is obscure to shop floor personnel, and there is no participation or learning in this approach; it is difficult for shopfloor supervisors to accept this because they traditionally considered a certain level of discretion with their operations schedules to be important. Also, as pointed out by [White 1990], OPT may be inappropriate to be used as a real-time scheduling tool where production is highly variable since it requires significant time to generate schedules. OPT is also weak in dealing with certain types of manufacturing phenomena which occurs during production, such as 'moving bottlenecks' [Roy 1993].

Since the advent of OPT, there has been considerable interest amongst both researchers and practitioners in alternative methods for development of finite capacity scheduling tools.

2.3 Recent trends to shopfloor scheduling

Shopfloor scheduling systems usually consist of a predictive and a reactive scheduling mechanism, which are respectively responsible for generation of detailed production plan and reacting to contingency during production.

2.3.1 Review of approaches for predictive scheduling

Predictive scheduling addresses the problems of preparation of detailed short term plans, which may include job release, job sequencing on workcentres, determination of lot sizes for each job, and determination of the start time (finish time) for each job on each machine on which it must be processed. In many applications, the predictive scheduling system is supposed to be connected to an MRP/MRP II system, and it is therefore able to use the information provided by that system, e.g. due dates, lot sizes and order release. In this case, the main task of the predictive scheduling system is finite capacity scheduling, i.e. creating a feasible schedule in a given planning time horizon according to the specified due dates and lot sizes.

Predictive schedules are created in a static environment and the robustness of the schedules are usually described by capacity and other allowances, i.e. 'time buffer' [Goldratt & Cox 1984].

In current industrial practice, predictive schedules, in most cases, are generated by human schedulers with or without computer support. Gantt charts can be used for small-sized two-dimensional problems. For larger and/or multi-dimensional problems, e.g. DRC, a computerised approach is usually used.

Computerised scheduling tools usually require some methods to model manufacturing systems, and there are two methods often used, logical and mathematical.

Due to the complexity and dynamic nature of manufacturing systems, recent work on the development of practical scheduling tools have favoured logical modelling rather than mathematical modelling.

There are two commonly used approaches in logical modelling, time based approach and constraint satisfaction.

Simulation, especially discrete event simulation, is the most commonly used time based approach, and has been found to be an useful tool for finite capacity scheduling. The power of simulation as a tool comes from its capability to describe a manufacturing system to a considerable level of detail. Many simulation based finite capacity scheduling tools have been developed and are commercially available, e.g. [Marriott 1994]. These tools usually have a set of dispatching rules for the user to choose in generation of the schedule, and provide a user-friendly interface for modelling, and some even have an interface to connect to an on-line computer so that it will be able to generate a production plan from current shop status information. Although a human scheduler is still required to make decisions, such as in selection of dispatching rules, the simulation tools allow them to concentrate on decision making on some key parameters, such as capacity requirements [Roy & Meikle 1995].

The constraint satisfaction approach is often used in AI-based scheduling systems. This approach sees all the requirements on resource and/or job as constraints. They include their heuristic rules in them, but they are much more than knowledge based systems. They are based on treatment of both constraints and heuristics. ISIS (Intelligent Scheduling and Information System) [Fox & Smith 1984] is the best known AI production scheduler of this kind.

The core of ISIS is a framework for incorporating a wide range of real-world constraints i.e. the major factors that have impacts on system performance and need to be taken into account, for production scheduling. [Fox & Smith 1984]

categorised these constraints as organisation goals, physical constraints, causal constraints, resource availability and preference constraints. The constraint satisfaction approach has been adopted by many researchers and practitioners in later AI-based scheduling studies [O'Grady & Lee 1988] [Bharadwaj et al 1994] [Elleby et al 1988].

Discrete event simulation is well established as an efficient tool for developing accurate models of manufacturing system; constraint satisfaction techniques, however, tend to have a high computational burden, and often many details of the manufacturing system may need to be omitted in this approach.

2.3.2 Reactive scheduling

Reactive scheduling addresses the problem of maintaining a schedule in a dynamic and stochastic world [Burke & Prosser 1989]. Reactive scheduling is used to describe the process of identifying the errors emerging over time in a working schedule, and maintaining and repairing the schedule in reaction to the changed situation.

Reactive scheduling can be classified as full-scale rescheduling, dynamic reactive scheduling and controlled reactive scheduling.

Full-scale rescheduling

Full-scale rescheduling is to react from deviation of the working schedule by throwing away the current schedule and creating a totally new one. The scheduling system has to do a full scale rescheduling each time a deviation has been detected. This strategy is not only highly inefficient, since the entire problem has to be resolved every time a change occurs, but also unnecessary for most disruptive situations because many errors could be corrected by minor changes to the

current schedule or can be self-correcting. Large scale revisions to a schedule may also create uncertainty and instability on the shop floor.

In most current practical applications, development of a new schedule is difficult due to the huge size of its solution space, and most schedule generation has to be aided by a human scheduler through a heuristic approach for reducing solution search space. Although simulation and other computerised facilities could be used for assistance, rescheduling all the remaining orders is still time consuming, and has to be done in a static mode.

Therefore, full scale rescheduling is quite expensive, time consuming, and disruptive to production, and it is usually used as a final resort.

Controlled reactive scheduling

Controlled reactive scheduling is reactive scheduling for control. This approach concentrates on schedule maintenance.

This approach is derived from research in control theory [Gershwin et al 1986], which advocates controlling the need for rescheduling by anticipating the various events which might occur in the environment and explicitly storing the resulting strategies in an efficient manner. This approach has also been referred to as plan caching by [Noronha et al 1991].

It is difficult, if not impossible, to explicitly represent all the uncertainties in the environment and to develop an universal reaction plan, but it is possible to develop control strategies to cope with a limited set of and commonly occurring disruptions in the manufacturing environment by using this approach.

Dynamic reactive scheduling

Dynamic reactive scheduling addresses the needs for the reactive scheduling mechanism to detect and react to the changing environment. This approach conducts reactive scheduling by detected changes, e.g. violation of due date requirement, without being concerned with the cause of the changes (such as machine breakdown). Dynamic reactive scheduling represents a tendency to develop a generic rescheduling method.

Dynamic reactive scheduling is different from full-scale rescheduling since it only updates part of the predictive schedule in the face of disruption, and differs from controlled reactive scheduling since it does not attempt to anticipate the disruptive events in its problem solving process.

For fast execution, dynamic reactive scheduling is often based on local information.

Approaches closely related to maintaining shop floor stability

There are a number of problems in reactive scheduling:

- a) there is no optimal solution, or it is difficult to find one,
- b) the problems happen ‘unexpectedly’ on the shop floor and the effects of an unpredicted event can be quite difficult to detect,
- c) the problems have to be solved during production time, i.e. real-time.

Since it is difficult to find analytical solutions, a heuristic approach is usually used in reactive scheduling.

A priority rule based approach is the conventional approach in reactive scheduling.

By this approach, a priority rule is used for sequencing or dispatching jobs at workcentres. This approach, though fast in execution, may not lead to a good global solution, because it only uses local information in decision making and has neither included the knowledge to deal with the various disruption events, nor taken into account interaction between each action. In this line of research, a variety of priority rules have been advanced and tested by means of simulation [Conway et al 1967], [Holloway & Nelson 1974].

AI based reactive scheduling represents another direction of research. The problem of responding to a changing environment has been often studied and categorised by AI researchers as a problem of replanning. Replanning is an approach used by a planner when a mismatch between the expected and actual state of the world has been recognised. In this context of scheduling problems, a few AI-based systems to deal with decisions in changing circumstance have been developed [Collinot 1989], [Smith & Ow 1990], [Bharadwaj et al 1994].

OPIS [Smith & Ow 1990] takes both an order-based and resource-based perspective of the scheduling task, being able to dynamically switch between the perspectives during schedule creation and revision. OPIS attempts to closely integrate the tasks of predictive and reactive scheduling, but viewing both predictive and reactive scheduling as an opportunistic process rather than that of schedule maintenance.

In the research reported by [Elleby et al 1988], a scheduling task was viewed as being predominantly one of maintenance rather than creation, and the same mechanisms were used for both predictive and reactive scheduling. [Burke & Prosser 1989] also studied predictive scheduling and reactive scheduling from a schedule maintenance viewpoint. In their research, a distributed asynchronous scheduler (DAS) has been developed for modelling hierarchical structure of production scheduling, from individual resources, e.g. shift leader, up to the factory scheduler who decides what work must be done; task of scheduling and maintaining schedules is

carried out through communication and negotiation across the decision making hierarchy.

Problems of schedule maintenance in these pieces of research were often described as determination and satisfaction of constraints, and heuristic approaches were employed to describe the knowledge to capture these constraints, to integrate constraints into a search process, to relax constraints when a conflict occurs, and to compare various solutions to the scheduling problem. But, these systems are essentially addressing the problem of dealing with uncertain situations from replanning (rescheduling) point of view, i.e. the reactive scheduling mechanisms in these systems tries to reschedule all the jobs as a result of disruptive event(s). They use an ‘opportunistic’ mechanism for solving the rescheduling problem and are not concerned with maintaining schedule stability.

There is another direction of research in AI based scheduling, which has concentrated on improving the effectiveness of the dispatching mechanism, and these systems are often seen as real-time shopfloor control tools since they are used as an intelligent tool either to assist shop floor personnel to carry out dispatching, or automatically provide dispatching decisions, e.g. [Walker & Miller 1986], [Ben-Arieh et al 1989], [Ranky 1988].

2.4 Predictive and reactive scheduling for promoting shopfloor stability

Maintaining shopfloor stability, as stated in section 2.2, is a goal for shopfloor scheduling as well as a strategy for production management. This strategy advocates using robust predictive schedules to control the need for full scale rescheduling, and maintaining the schedule by developing specific strategies to deal with disruptions.

By this strategy, the shopfloor control system attempts to maintain original plan in the face of disturbances, without resorting to full-scale rescheduling every time some deviation occurs. The goals for schedule maintenance should be laid down in a predictive schedule, such as jobs' due dates, capacity plan, and so on.

In general, maintenance of shopfloor stability needs a “good” predictive schedule which is efficient, robust and has clear objectives for control, and a good controlled reactive scheduling mechanism to carry out schedule maintenance during its execution.

2.4.1 Schedule maintenance and schedule repair

Schedule maintenance involves all the activities for maintaining a predictive schedule, including identification of sources of disturbances, assessment of the extent of deviation of production from a predictive schedule, and devising corrective plans. The objective of schedule maintenance is always to bring the production process back to the predictive schedule as far as possible.

Schedule repair involves the activities for repairing a schedule on recognition of having some errors in the current plan, e.g. late orders. Schedule repair emphasises the necessity of schedule modification, and objective of schedule repair may not be strict maintenance of shopfloor stability.

Schedule maintenance would need to carry out schedule repairs when modification to the current schedule is necessary, but aims to maintain the current schedule, i.e. to make the repaired schedule track the original schedule well.

Schedule maintenance involves broader control activities for maintaining a predictive schedule through manipulation of the schedule itself, e.g. job resequencing in a queue, to manipulation of the manufacturing environment such as use of overtime work.

2.4.2 Problems and approaches in maintaining shopfloor stability

Maintaining shop floor stability is concerned with the global performance of a scheduling system which involves predictive scheduling and reactive scheduling.

The general problem of maintaining shopfloor stability can be described as follows: Given a manufacturing environment along with the information to manage the environment, information on delivery plan and current schedule status, information on disturbances to the delivery plan and about deviation from it due to changes in the manufacturing environment, modify the current schedule so as to take the deviation into account. This modification has to be done such that the impact of the deviation on the rest of the schedule is the minimum possible. In other words, the aim is to develop a new schedule which could meet promised delivery dates and get the shop-floor back to its target with minimum cost.

For maintaining shopfloor stability, the predictive scheduling mechanism needs to generate a schedule which is efficient in terms of meeting due dates and production costs, and robust for coping with uncertainty. Predictive schedules also need to have a detailed timing plan as objectives for schedule maintenance in real-time environment during its execution. Operation target start and completion times are required as intermediate objectives to monitor extent of deviation of production from its plan, and decide corrective actions for schedule recovery.

As described in chapter 1, a predictive schedule can be seen as a guideline for reactive scheduling. In other words, in addition to other constraints, e.g. physical constraints on the shop floor, the plan itself represents some of the constraints which need to be satisfied in reactive scheduling.

The available time to carry out the task might be one of the most serious types of

constraints to schedule maintenance, since reactive scheduling needs to be carried out in real-time.

For this reason, a combined (controlled and dynamic) reactive scheduling strategy appears to be appropriate for maintaining shopfloor stability. Controlled reactive scheduling allows disruptive event and/or situation related knowledge and intuition (gained through first-hand experience by the shop floor personnel) to be included, while the approach of dynamic reactive scheduling can be used for detecting and taking into account the changes in the manufacturing system.

Chapter 3

Review of approaches to shop floor control

Shop floor control is the lowest layer of a production management system and governs very short-term detailed planning, execution, and monitoring activities required to convert released orders by the planning system into a set of completed orders.

Like all other control systems, shop floor control systems involve setting of objectives (targets or goals), devising methods for measuring performance against the objectives, evaluating performance (especially deviations from the planned results), and deciding on corrective actions (where this is possible) to get back on schedule.

3.1 Objectives of shop floor control

3.1.1 Classification of objectives of shopfloor control

Shop floor control systems can be seen as an execution of long term plans produced by MPS and material planning, so the ultimate objective of shop floor

control is to achieve the plans in a cost efficient manner.

Objectives of shop floor control can be divided into the following four categories concerning the control of WIP (Work In Progress), quality, labour and equipment [Browne et al 1988]:

- 1) **WIP** - reduced WIP investment, balanced workload, improved delivery performance, and reduced manufacturing lead time.
- 2) **Quality** - reduced incidence of defects and scrap.
- 3) **Labour** - improved efficiency, improved utilisation, increased operator satisfaction.
- 4) **Equipment** - improved utilisation, improved availability and reduced set-up costs.

The goal of shop floor control can also be related to operation related goals, e.g. machine utilisation, and customer satisfaction (such as meeting due dates, expected quality and quantity). In a competitive market, there is an increasing emphasis on customer satisfaction; failure to meet customers' requirements can result in delay penalties, and/or loss of goodwill due to possible damaged reputation. But, failure to control production cost could also affect the company's ability to compete.

3.1.2 Maintaining shop floor stability

In job shop and discrete batch production, if predictive scheduling is used for generating short term schedule(s), then (real-time) control of production can be based on the predictive schedule(s) with or without considering other goal functions. Predictive scheduling and real-time control, when used together, usually have different goals; predictive scheduling aims to generate a 'good' schedule, and

real-time control tries to make production follow the schedule and in this sense, the schedule itself is a goal for real-time control. Since the objectives of scheduling are usually interrelated and in some cases conflicting, appropriate coordination of them is important for shop floor control in both predictive scheduling and real-time control.

Maintaining shop floor stability emphasises the need for the production process to adhere to the predictive schedule, i.e. the predictive schedule itself should be the goal of control during production as long as the schedule is feasible. It has been stated [Roy 1993] that robustness of the schedule itself should be a goal for predictive scheduling in addition to other goals, since it could smooth some disturbances and provide the shop floor control system with the greatest chance to recover from any deviation that may occur due to random events (e.g. machine breakdown).

Maintaining shop floor stability is based on the assumption that the predictive schedule is good (even best) and constructed from appropriate coordination of all the goal functions. Under the goal of maintaining shop floor stability, the shop floor control system is encouraged to control a production process according to intermediate objectives laid in its predictive plan, such as operation due start times. These intermediate objectives represent the detailed resource allocation plan and the resource capacity plan. When there is no occurrence of disruptive events on the shop floor, following the plan will guarantee achievement of the production schedule without extra cost. After occurrence of a disruptive event, the production process may deviate from the plan, but, in many situations, especially in situations where there is no imminent risk of violation of due date objectives, these intermediate objectives can still be used as guidelines for effective shop floor control.

The central issue for maintaining shop floor stability is control. If the shopfloor is out of control, which means that the company is unable to consistently deliver

a quality product at the time, price and place promised, variability in the supply chain (internal and external) increases and leads to longer lead time and, hence, cost. Frequent changes in production plan will also lead to instability on the shopfloor, and reduce its efficiency. From this viewpoint, maintaining shop floor stability is an important, combined goal to improve efficiency and predictability of production management systems.

For maintaining shop floor stability, it is important to decide how and in what way to relate production to the original plan. The main concern in this issue is if and to what extent the plan will be maintained when production process has deviated from the plan and by what control functions.

3.2 Major control functions in shop floor control systems

The problem of shop floor control can be addressed from two dimensions: control of progress of production, and control of the manufacturing environment. The former addresses the problems in control of progress of orders on the shop floor, and the latter addresses the problems in controlling environmental related factors, e.g. availability of resources.

3.2.1 Functions for control of manufacturing environment

Control of manufacturing environment aims to control and/or adjust some controllable environmental variables, such as resource availability and capacity, resource requirement, and production process related variables, etc, to cope with disturbances in the environment. There are normally two kinds of control functions in this category: one can be referred to as long term strategy; the other as short term.

Manufacturing companies usually have some long term strategies to specify the control methods and means which can be used to cope with the most likely occurring disturbances. It usually includes facility maintenance, and quality control.

Facility maintenance

Facility maintenance aims to guarantee that all the production facilities in a manufacturing environment are in good or 'best' working condition according to their specification. Facility maintenance can be preventive and/or reactive, which are respectively for reducing the frequency of equipment failure, and reducing the severity of equipment failures once they occur. In job shop and discrete batch production, preventive maintenance can be scheduled before production, and carried out without interference to production. But, due to the complexity of the production process in such environments, breakdown maintenance is often required and, in most cases, carried out by specially trained maintenance worker(s).

Quality control

Quality control aims to guarantee and improve quality of products, and reduce incidence of rejected items and necessity for rework (both of which affect capacity), by control over process related parameters. Consistency, degree of precision in the process, and extending its performance frontier are usually identified as the process capabilities within this framework.

In industrial manufacturing environments, objective and strategy on quality control are influenced by the manufacturing environment itself (production process and product positioning), and the manufacturing philosophy adopted by management. In repetitive manufacturing environments, since workflow pattern is simple, it is relatively easier for management to trace sources of product defects and variations in product quality. In such environments, JIT philosophy and

Total Quality Control strategy, which aims to seek zero defects in production (i.e. producing parts 'right the first time'), has been successfully used. In many job shop and discrete batch production, however, complexity of the production process makes it more difficult for management to make long and continuous efforts to trace the sources of variation in the production process in a cost efficient manner; therefore they often prefer to react and control abnormal variations, i.e. returning the process, whose characteristics falls outside the acceptable range, back to the acceptable limits rather than consistently pursue elimination of the sources of variation.

Capacity control functions

Capacity control usually refers to the approaches which can be used to temporarily adjust resource capacity and/or requirements on the shop floor in order to compensate for any difficulties being experienced to meet planned production progress, and/or substantial capacity losses during production. Commonly used capacity control approaches include:

- Spare capacity.
- Changes in work rate.
- Use of overtime.
- Subcontracting.
- Alternative routings.
- Lot splitting.

Some of these approaches, e.g. spare capacity, spare parts inventory and subcontracting arrangements are often seen as long term (and medium term) strategies to provide the shop floor with a degree of protection from the effect of disturbances.

The others are usually seen as short term means to cope with disturbances on the shop floor.

A certain level of spare capacity is necessary for smoothing work load among resources, and this approach is sometimes referred to as robust scheduling, by which some capacity cushions ('time buffers') can be planned when preparing work schedules. Such 'time buffers' are more cost efficient than standby machines, and this approach is advocated by OPT and also suggested for maintaining shop floor stability [Roy 1993].

These capacity control functions manipulate a manufacturing environment in two different ways. Some of the approaches change total resource capacity or requirements in the planned production time horizon, e.g. use of overtime can temporarily increase resource capacity, and subcontracting could reduce resource requirements; the others would change capacity requirement on specific resource in a particular time span and/or at a particular time without affecting total resource capacity and requirements, e.g. lot splitting.

Capacity control often leads to increases in cost in one way or another. Control of a manufacturing environment by changing total resource capacity or requirement would usually lead to increases on investment or production costs; while control by lot splitting and alternative routings, has other associated costs such as lot splitting is at expense of extra setups.

3.2.2 Major functions for control of progress of production

As described by [Bertrand & Wortmann 1981], there are two basic control functions for control of progress of production in shop floor control systems, order release/review (ORR) and detailed assignment.

ORR is the link between planning and execution, which controls the flow of orders from the planning system to the shop floor. It is responsible for determining what orders are to be released to the shop floor, at what time, under what conditions. For approval of job release, inventory status of components and raw materials required by the order need to be checked, as well as capacity availability in the system.

ORR is usually seen as a filtering mechanism and a capacity management tool to smooth out peaks and valleys of load on both the shop floor as a whole and on various work centres.

After a job has been released, it would be added to the set of released, unfinished jobs waiting for execution. Job release is, thus, the major control activity before execution.

Detailed assignment is the function to decide allocation of production capacity for execution of remaining operations of the released, unfinished jobs on the shop floor. Capacity may refer to operator capacity as well as to equipment capacity. An operation of a job can only be processed when all its required resources are available. After execution of an operation, the job returns to the set of released unfinished jobs - unless the job is finished.

Dispatching is the major activity in detailed assignment and responsible for selection of the next job to be processed and the corresponding assignment of resources, e.g. workers and tools to the selected job. When there are multiple resources needed for jobs, their capacity allocations need to be synchronised.

3.2.3 Interaction and relationships between different type of control functions

In general, the functions to control progress of production are often seen as the basic control functions in shop floor control systems since they directly manage work flow on the shop floor; while control of manufacturing environments affect capability of shop floor control (long term functions) or are used when there are difficulties for the basic control functions to control production progress to meet its objectives (short term functions). Usage of short term approaches to controlling the manufacturing environment are usually restricted by consideration of cost. For example, since overtime cost rates is frequently greater than that for regular time work, companies usually have a strategy to limit its usage by placing some cost related constraints, e.g. limiting the number of hours of overtime work hours per day.

Between dispatching and ORR, the two basic control functions, dispatching is usually seen as the major control activity in shop floor control, but in recent years there are some reported research work on studying the functions and conditions of ORR in improving system performance [Melnik et al 1992], [Melnik et al 1994], [Park et al 1995].

Review of research on ORR control strategy

As one of the basic control functions in a shop floor control system, ORR is expected to play an important role to improve performance of the manufacturing system. By controlling flow of work to the shop floor, ensuring that the shop is not overloaded and helping to smooth loading peak and valley at workcentres, ORR, conceptually, should have a positive impact on the operation of the shop floor. And, when ORR works effectively, as shown by some researchers [Melnik et al 1994], dispatching could be simplified. But, one of the important

findings in these research is “the performance of an ORR system is strongly dependent on the presence of variance control at both the planning and shop floor levels.” [Melnik et al 1994]

On the shop floor, variance control refers to controlling the extent to which data describing measurements of interest (e.g. number of jobs in queue over time, processing times at the various work stations, shop load) are spread out or dispersed about its mean. Some of the major findings in these research [Melnik et al 1992] [Melnik et al 1994] are summarised as follows:

- ORR is not a general or broad based tactic since ORR does not work well under all conditions. Effectiveness of an ORR system is strongly dependent on the presence of variance control at both the planning and shop floor levels. It is found that ORR has a specific range of variances in which it operates most effectively, such as the situations described by [Melnik et al 1992].
- Variance control affects not only effectiveness of ORR but also the importance of the dispatching process. When variance control is present, concern over dispatching efficiency (i.e. which dispatching rule is used) becomes far less important. Otherwise, when variance control is not present either in the planning system or on the shop floor, selection of the most appropriate dispatching strategy or rule becomes very critical.

In job shop and discrete batch environments, it is difficult to control the system variance within the levels assumed in the above mentioned research because of the uncertainty in the environment. Especially, an event of machine breakdown could interrupt processing of an order, and impose changes to the completion time of its current operation. And as observed by [McKey et al 1988], “When an unexpected event occurs, its effect normally lasts longer than the batch processing time for the work process in the area affected”. When variances (described by variation of operation processing times) on the shop floor are great, as reported

by [Melnik et al 1992], ORR is overwhelmed and cannot adequately respond to the large changes on the shop floor.

In addition to variance control, there are other factors which could limit effectiveness of the ORR function in shop floor control. Effectiveness of ORR can be limited by the feedback mechanism since, in many real manufacturing environments, information about shop status can only be fed back to an ORR system after a certain time interval, e.g. once a week. When ORR is used to smooth the work load level of a shop where there is a bottleneck, its effectiveness can be influenced by the position of the bottleneck [Park et al 1995].

Although there are many ways to control variances on the shop floor, in job shop and discrete batch environments, however, complexity of the production process and dynamic nature of the manufacturing environment make it difficult, if not impossible, to control the system variance within such levels at which dispatching becomes unimportant.

3.3 Selection of shop-floor control actions

Control actions are the corrective activities required in reaction to changes in circumstances on the shop floor during production. Control actions can take place at different levels of a manufacturing system (equipment level, cell level and system level). In production management systems, shop floor control actions usually refer to the activities at cell and system levels, which do not directly control machine processing and part movement but make decisions to control the activities at its equipment level. Control actions in this category are various, as briefed in the last section, ranging from reallocation of resources to actions for capacity control.

3.3.1 Various control actions

In a real manufacturing environment, it is very common that a job can obtain resources ahead or behind its scheduled time in the work plan. In this situation, timing of capacity allocation for the job may need to change, and a decision needs to be made about if and when this job should be allowed to obtain the required resources for setting up and processing, i.e. reallocation of resource capacity for the job.

This action is the most frequently used action on the shop floor since it can prevent loss of resource capacity when a job has been delayed. In job shop and discrete batch production environments, queuing time is the largest part of the overall lead time; therefore, a delayed job is often likely to have a chance to catch up without adjusting related resource capacities, if queuing times at its downstream workcentres are reduced.

By this action, only timing of capacity allocation for all or some of the remaining operations of the job needs to be changed, and there is no change on total capacity requirements for each workcentre and each type of worker, and routing and batch size of the job.

Other actions are referred to as capacity control actions since they could change resource capacity or requirement in one way or another. Capacity control actions usually refer to the short term means to adjust resource capacity or requirements. As discussed in section 3.2.2, overtime is one of the most commonly used approaches since in many discrete parts production environments, it is more reliable than subcontracting and more cost efficient than spare parts inventory. Some modern production management strategies, e.g. OPT also advocate using overtime as a basic means for capacity management. [Goldratt & Cox 1984]

Feasibility of control actions are determined by a variety of factors or constraints.

3.3.2 Important factors in selection of control actions

The most important factors in selection of control action are its objectives i.e. control actions used must be consistent with the predetermined goals.

For maximising machine and/or system utilisation, alternative routing is often used for control of FMS systems to take advantage of their flexibility. In many manufacturing systems, however, job routing and lot sizes are determined by the planning system and can not be allowed or are difficult to be changed on the shop floor. For maintaining shop floor stability, therefore, the routings and lot sizes decided by the planning system should be respected.

The feasibility of control actions is also constrained by other factors. For example, changing equipment work rate is restricted by types of equipment and their current working conditions; alternative routings by the production process, flexibility of machine, etc; using overtime is constrained by the shift systems and labour availability. Effectiveness of some actions are constrained by shop floor situations, for example, lot splitting "is useful only when a portion of the original shop order is required. If the entire lot is required, this option offers no advantage." [Melnik et al 1985] Many actions have some side-effects on the production it controls, e.g. lot splitting will create additional setups.

By taking these constraints into consideration, job reallocation and overtime are usually the most useful control actions in job shop and discrete batch production. They are usually used together in industrial manufacturing environments.

The main task of the control actions is to respond to disturbances in manufacturing environments. When corrective actions are required, the control system must make a decision, which is a combination of available corrective actions based on information on current shop status and knowledge of the disturbance, e.g. type of disruptive event, stage of its occurrence and so on.

3.4 Disturbances and resource flexibility

Disturbances to production are usually caused by unexpected events in the manufacturing environment. Uncertainty in a manufacturing environment could be due to various reasons, such as equipment failure, labour absence, or because the manufacturing system in question is not understood in sufficient detail, production and control are based on incomplete information, or even measurement errors caused by malfunctioning instruments.

Since manufacturing companies vary greatly in their products, and production processes, the uncertainty and the requirement for shop-floor control would vary from one type of production environment to another. [Melnik et al 1985] emphasised that, these differences affect the design of appropriate shop floor control systems or at least the features of these systems e.g. importance and timing of each control activity.

3.4.1 Disturbances in job shop and discrete batch manufacturing

In a job shop or batch production, process oriented approach is usually used, e.g. using general purpose machines and function oriented equipment layout. In such an environment, with its constantly changing jobs and varying processing requirements, the shopfloor control system must be able to handle a large amount of uncertainty.

For analysing effects of disruptive events, the following major characteristics need to be studied:

- a) Types of disruptive events
- b) System components associated with the disruptive event

- c) Stage of its occurrence
- d) Recovery time
- e) Frequency of occurrence
- f) Impact on the system

Disturbances can be caused by a variety of unexpected events, [Farhoodi 1990] classified these disturbances as resource related, order related, material related, information and operation related.

The common disruptive events which happen in a manufacturing environment are:

- a) machine breakdown,
- b) rush job and job priority changes,
- c) labour absenteeism,
- d) late arrival of material.

Different types of disturbances have different impact on different components of a production system, but a disturbance will directly or indirectly affect usage of resources.

Some disturbances could impose change either on resource requirement or resource capacity in the planning time horizon, e.g. machine breakdown will reduce total available capacity of that machine and its work centre; variation of part process time will increase or decrease relevant resource requirement. Some will enforce changes on timing of resource assignment, e.g. rush job may affect other jobs' timing to obtain resources.

A disturbance could be propagated through sharable production resources in the manufacturing system through a series of violation of time constraints defined in the predictive schedule. The method of propagation is determined by type of the disruption event. For an order related disruption event, e.g. rush job, the propagation is by way of resources required by its uncompleted operations. For a resource related disruption event, such as machine breakdown which would reduce the machine's available capacity, it could affect utilisation of downstream machines and result in loss of other machine's capacity. An effective shop floor control system should have strategies to prevent the propagation or to limit the effects of the propagation.

It is difficult to predict the impact of a disturbance due to propagation. This is not only because of the limited time available to the shop floor to make such an assessment, but also because of the need for complete and accurate information. Even if all the information was available, errors may be caused by a mismatch of information overload and limited human ability in information processing [Prabhu et al 1992].

The extent of the impact caused by a disturbance is influenced by a variety of factors, including quality of the predictive schedule, type of the disruptive event, and resource structure of the manufacturing system.

3.4.2 Resource flexibility for dealing with disturbances

The structure of production resources are characterised by type, quality and quantity of resources, and the way they are organised. The capability of a manufacturing system for dealing with disturbances is determined by its resource structure.

Resource structure includes equipment structure and human resource structure.

Structure of equipment resources is usually described by the production process and human resource structure described by organisational structure.

There are two main sources of resource flexibility, worker flexibility and machine flexibility. Machine flexibility can be defined as the number of different items which can be processed by a (or group of) machine(s); worker flexibility can be defined as the number of workstations or machines that a worker can operate, and/or types of operations which can be processed by a worker [Malhotra & Ritzman 1990].

Flexibility of individual resources forms the basis for achieving a higher level of flexibility in a manufacturing system, e.g. routing flexibility. A higher level of flexibility, however, is determined by a variety of factors in a manufacturing environment, such as worker related flexibilities which can be influenced by the organisational structure as well as workers' training.

As reported by [Malhotra & Ritzman 1990], resource flexibility is important to shop floor control and "is particularly important when there are disruptive variabilities and uncertainties in the manufacturing environment." since it "can reduce the need for generous capacity cushions, without affecting performance adversely" [Malhotra & Ritzman 1990].

Automation has long been used for improvement of machine flexibility. Advanced manufacturing systems, such as Flexible Manufacturing System (FMS), can incorporate different automation concepts into one system so as to provide a variety of flexibility through integrated automation technology, but these flexibilities require high investment cost. Therefore, only a small amount of work shops have been equipped with such systems in manufacturing industries.

In discrete parts manufacturing, worker flexibility is important in dealing with disturbances. As shown by [Sheu & Krajewsk 1994], worker flexibility has similar

importance to machine flexibility for improvement of system performance in a dynamic environment. And, in many manufacturing companies, there is always some worker flexibility. As reported by [Sharma 1987], on average, workers can work on 5.4 work stations. In an industrial environment, it is common to use worker flexibility to cope with disruptive events on the shop floor, such as when a machine is broken down, the worker working on the machine could be reassigned to a downstream machine to prevent possible job congestion on that machine.

3.5 Approaches and systems for real-time control

The problem of shop floor control has been studied by researchers in operations research in terms of dispatching. Such work have attempted to provide the dispatcher with rules for determining job priorities and to identify the conditions (in terms of such factors as shop load, due date tightness, etc) which affect the operation of the various dispatching rules. They have often aimed to improve the process by which job priorities are determined and assigned.

3.5.1 Priority rule based dispatching

Dispatching rules are used primarily to help shop floor personnel (manager, dispatcher, operator, etc) to make decisions about job selection and sequencing on the shop floor.

Priority rules can be classified from different dimensions, and [Gere 1966] gave basic definitions of rules and classified priority rules from three dimensions: environment, complexity and scope. By these dimensions, priority rules can be either static or dynamic; simple or complex; local or global. By his definitions, a priority rule is static if either priority of each operation has been assigned before

production and is not going to be changed during execution, or varies in a way independent of the schedule and current job status; it is simple if the data concerning current job status has not been used; it is local if only information on the dispatching workcentre is used in a priority function of the rule. A priority can be described by the first two dimensions which are based on the priority function of the rule, e.g. EDD is referred to as static since only order due dates (which are unlikely to be changed during production) are used; some machine related rules like WINQ (select job for which the direct successor workcentre has the least work load) and NINQ (select job for which direct successor workcentre has the shortest queue) are seen as global rules since they use information on other workcentres. But, scope of a priority rule is sometimes related to its manufacturing environment and/or structure of its dispatching system. For example, in a manufacturing environment with single sharable resource, EDD is referred to as a local rule; while in a DRC environment, EDD can be global since information (order due dates) on more than one workcentre may be used for job selection. In DRC environments, the rules which use current shop status information are usually seen as global dynamic rules, e.g. CRR (Critical Ratios), WINQ and NINQ.

Priority rules can also be classified according to the objective function used [Panwalker 1977], [Gupta et al 1989], such as due date related rules, machine related rules, etc.

For acceptance of dispatching rules, performance of these rules must be demonstrated against the criterion or criteria which the shop floor personnels perceive as being the most relevant.

Since shop floor control is part of a production management system, the planning system determines the type of dispatching rules to be used. In MRP based planning system, only due-date-based dispatching logic is appropriate because non-due-date-based rules, such as SPT or FCFS, are not consistent with the

due-date orientation of MRP. Most commonly used dispatching rules include [Kanet & Hayya 1982]:

- 1) Earliest due date (EDD)
- 2) Minimum job slack (Slack)
- 3) Critical ratio (CR)
- 4) Earliest operation due date (ODD)
- 5) Minimum operation slack (S/OPN)
- 6) Operation critical ratio (OPCR)

The first three dispatching rules are job-based, and the final three are similar to the first three, but are operation-based. The operation-based rules are intuitively appealing because they provide intermediate benchmarks for job progress, and [Baker 1984] and [Vepsalainen et al 1987] also show the superiority of decomposing job due dates into operation milestones, and using these operation due dates (ODD) for setting priorities. Of three operation-based rules, ODD is increasingly becoming the rule of choice for due date driven systems since it not only provides a milestone about job progress, but also states the priorities in terms meaningful to most shop floor personnel.

But, since a due date based dispatching rule does not include current information, “most managers working in discrete batch environments were of the opinion that order sequencing was best done by humans and not by computerised dispatching algorithms. The priorities generated by the dispatching rule were only inputs to the dispatching process. They felt that no dispatching rule could adequately consider all of the relevant factors.” [Melnik et al 1987] In other words, shop floor personnel usually play an important role in shop floor control.

This encourages use of knowledge based approach in shop floor control.

3.5.2 Knowledge based approach for Shop Floor Control

As described in the last section, it is often the responsibility of shop floor personnel to cope with exceptional situations, but, decisions which are made manually without the aid of well-coordinated information are less than satisfactory (especially in data intensive manufacturing environments, such as job shop and batch production.)

In contrast, expert systems can capture the heuristics of the shop floor personnel and coordinate their knowledge with quantitative data in computer usable formats and allow the user to arrive at a 'good' decision in a limited time.

Such a system can be used for shop floor control to either automatically make decisions or to assist shop floor personnel in decision making. The systems for this area of applications can therefore be categorised as two types: one is for automation of the control process, and the other is for providing intelligent help to shop floor personnel.

Knowledge based FMS control systems are typical examples of the first type of applications, e.g. systems developed by [O'Grady & Lee 1988]; [Wu & Wysk 1989]; [Ben-Arieh et al 1989]; [Walker & Miller 1986]. In such a system, the functions of scheduling and control are often integrated, i.e. a job is scheduled when it arrives at a workcentre of the system, and there is no predictive schedule, or it would not be followed. The control system does not attempt to maintain shop floor stability, and objectives of control is usually for maximisation of machine or system utilisation. Such an approach is totally reactive in nature.

[Ben-Arieh et al 1989] developed a knowledge based system for controlling a FMS cell at cell level and equipment level (robot movement). At the cell level, they used a look-ahead mechanism to project current shop status and detect the conflicts among the operations in the future, so that job selection could be based on

both current shop status and some information on future shop status. Disruption of machine breakdown was taken into account in the research. A knowledge based dispatching system (DISPATCHER) was implemented by [Walker & Miller 1986] to dispatch the orders on an automated network consisting of storage, transportation equipments (robots and conveyers), and workstations. The DISPATCHER maintains a database of products and their manufacturing processes, and alternative manufacturing operations at each workstations. In response to assignment requests from workstations, the system selects an order by considering the alternative operations which the workstation is currently able to perform, the priority and due date of these orders, as well as the overhead of any possible storage operations. [Wu & Wysk 1989] developed a multipass expert system for scheduling FMS cell (MPECS). Instead of using a knowledge based system to carry out dispatching, i.e. selecting the next operation, the knowledge based system was used for selecting a dispatching rule from a set of priority rules; a simulation module in the system was used to evaluate the alternatives in dispatching rule selection. An intelligent cell control system (ICCS) has been implemented by [O'Grady & Lee 1988] for supporting real time control on a FMS cell. The system consists of four blackboards that are respectively responsible for scheduling, dispatching, monitoring and error analysing.

Research on the second category aims to provide a decision support environment, in which shop floor personnel (planners, dispatchers and operators) can efficiently generate, explore and compare alternative schedules, as well as examine the effectiveness of corrective actions in different situations. This approach attempts to provide an intelligent tool to assist the planner to obtain and accumulate knowledge of potential interactions among corrective actions.

The system called LOGICA developed by [Farhoodi 1990] is a typical example of this type of system. LOGICA combined conventional (simulation, network, etc) schedule generation tools with knowledge-based schedule evaluation and repair. In the system, a generated schedule can be evaluated according to a variety of

criteria, e.g. job lateness, workcentre overload (underload), throughput time, etc; and improved until a satisfactory schedule is produced. A schedule repair subsystem in the system is available for suggesting or testing repairing actions in relation to the existing schedule, and executing corrective actions based on either short-term measures which aims to quickly recover from disturbances, or medium to long term measures for preventing a future schedule from becoming invalid. Both the schedule improvement and schedule repair systems have the capability of computing the potential contribution from each action as well as the capability to determine the most appropriate combination of actions. But, the LOGICA aimed to provide an interactive and user friendly tool to assist users to develop some control strategies by themselves; it has not included control strategies to cope with disturbances for achieving particular goals for shop floor control, e.g. maintaining shop floor stability.

Some of the knowledge based systems were developed for specific systems and therefore detailed knowledge about the system and the manufacturing environment can be used in the control, e.g. [Ben-Arieh et al 1989] used known machine repair time in his study. In other applications, such as systems developed by [Farhoodi 1990]; [Pluym 1990], knowledge based systems are relatively independent of a specific environment. In such a generic system, no reference is made to particular machines within a cell; they are only defined parametrically. In this way no changes need to be made when, for example, a new machine is added to the cell. By making small adaptations to the knowledge base, it is still possible to fine tune the system to particular characteristics or demands of the environment by modification of the evaluation criterion.

A generic problem decomposition approach has often been used in some applications [Ranky 1988]; [Pluym 1990]. This approach focuses on detecting conflicts among the operations for accessing production resources, which are caused by disruptive events, and then searching for solutions to minimise the effects of the conflicts. Due to the limited time available for decision making in real time con-

trol, this approach often turns to seek local optimal solutions. Control by this approach is essentially based on the consequences of disruption events, rather than disruptive event based problem categorisation. Other applications are based on problem categorisation, e.g. [Sarin et al 1990], i.e. each of the problem categories has an associated knowledge source with its own problem solving structure. As observed by [Sarin et al 1990], for many types of problems in a factory, shop floor personnel have a unique set of problem solving strategies. This approach is more effective and efficient than adopting generic problem decomposition approach in many situations for real-time control. But, when a number of different disruption types occur at the same time, it is difficult to aggregate actions based on each problem category. Therefore, including both of the approaches in a knowledge based shop floor system is the best way to achieve efficiency and generality.

[Farhoodi 1990] identified two trends in the development of AI-based planning and scheduling systems; the first is a trend away from automatic plan generation towards providing intelligent help to the planner, and the other is a trend away from general purpose systems towards problem specific systems incorporating greater knowledge about the domain.

Chapter 4

Control strategies for maintaining shop floor stability

Every shop floor control system has some strategies which provide guidelines to conduct activities on the shop floor. Since dispatching is one of the most important control functions in a shop floor control system, effectiveness of dispatching strategies is crucial to improve performance of the manufacturing system.

4.1 Dispatching strategies

4.1.1 Problems with simple dispatching rules

As discussed in 3.5.1, in many industrial environments, a due date based priority rule, e.g. ODD, is often used to rank the waiting orders and seen as a formal dispatching strategy. The dispatching rule, although it can select the next job by satisfying the predetermined criteria, is unable to capture the dynamics of the real problems involved, such as capacity problem caused by a disruptive event. And during production, there is little opportunity to refine, restate, or change this rule in a timely manner on a minute-by-minute or day-by-day basis. Therefore, in real manufacturing environments, the job sequence generated by the dispatching rule

is only regarded as recommendations because “no priority rule can consider all of the factors that affect the ultimate sequence” [Melnyk 1988]. In an industrial environment, shop floor personnel often play a key role in job selection.

In industrial environments, a dispatching process could involve three groups of people: the dispatchers, the department supervisors, and the operators. In effective shop floor control systems, observed by [Melnyk 1988], it is these people who bring important insights and information to the dispatching process, and manipulate the priorities of orders and determine the exact sequence in which orders are to be processed, and such information is a primary source of shop floor efficiency. But, as pointed out by [Melnyk 1988], the source of this information (often very qualitative) has not been considered by the dispatching procedures.

Therefore, as suggested by [Melnyk 1988], there is a need for research to modify existing dispatching strategy or rules to provide a means of using such information. And this approach, as stated by [Melnyk 1988], represents a direction of research that “pushes research into global-based dispatching rules - a direction generally scorned in past research studies.”

In other words, there is a need for dispatching procedures to include control strategies which are not currently included in research literature, but are frequently used by shop floor personnel and prove to be effective in the practice of shop floor control, so that such a real time control mechanism could take advantage of a broader and strategic view in its decision making. For this, it is essential to identify and describe the factors, i.e. constraints, which have or will have significant effects on performance of the production systems at the time of dispatching.

4.1.2 Major factors or constraints for consideration in dispatching

[Fox & Smith 1984] categorised the constraints for the problem of scheduling as:

Organisational goals - the goals which reflect global concerns and objectives of the company and imply general criteria against which prospective schedules can be compared. Goals and objectives are always seen as important constraints in dispatching, but are often conflicting. As described by [Melnik 1988], a dispatching mechanism should aim to meet at least the following three objectives:

- 1) To ensure that orders can be completed by their due dates.
- 2) To improve throughput of the manufacturing system.
- 3) To help maintain a constant load across work centres on the shop floor.

For maintaining shopfloor stability, meeting due dates may conflict with the objective of minimising operation costs, e.g. overtime cost.

Physical constraints The constraints which define the functional limitations of specific resources on the shop floor, such as a machine can only operate on a particular class of parts.

Causal constraints The constraints which establish precedence relationships and dictate the conditions for initiating an operation, such as the sequence of operations to produce a part and operation resource requirement for specific time periods.

Resource availability The constraints which refer to the resource availability constraints declared in an established schedule, i.e. when a resource gets assigned to a task for a given time period, the same resource can not be assigned to another task during the same time period. Such constraints can also be imposed by events which occur randomly, e.g. machine breakdown.

Preference constraints These are 'soft' constraints expressing preferred choices among alternatives in decision making, and reflect the heuristic knowledge

present in a given environment such as job priority, worker's preferences for certain shifts, etc. Preference constraints are considered implicitly when preparing the initial schedule. However, for control of schedule execution, the preference constraints may have to be violated to ensure feasibility of the schedule, e.g. a worker may have to do overtime work he does not like.

Due to the dynamic nature of the manufacturing environment, these constraints will affect production differently from time to time during schedule execution, and the major task of a dispatching strategy is to identify the constraints which have significant influence on outcome of the control actions at the time of dispatching. Since many constraints, e.g. resource unavailability, are caused by disruptive events, some major characteristics of disruptive events need to be taken into account in the design of control strategies.

4.2 Disruptive events

Disruptive events are those activities that take place at irregular (random) intervals and have a disruptive effect on the manufacturing process. To design effective control activities, it is crucial to understand some major characteristics of these events.

4.2.1 Major characteristics of disruptive events

As described in chapter 3, major characteristics of a disruptive event include the type of event, time and position of its occurrence, stage of occurrence, recovery time, frequency of occurrence, and its impact on the system.

Unpredictability is a common characteristics of all types of disruptive events. They occur randomly, and can not be predicted with any degree of reliability.

Time of occurrence, event type and position, and other parameters of a disruptive event can not be foreseen until it has actually occurred. Even if statistical data about a certain type of disruptive event is known, e.g. given statistical data about a machine's breakdown intervals, there is no way to know in advance precise information about this type of event, such as the exact time when this type of disruption is going to occur.

Disruptions reduce the available productive capacity of a resource, but not all disruptions have the same impact on production flow and system performance. Types of disruptive events are usually categorised by source of the events. There are many types of disruptive events on the shop floor, and the most commonly encountered events are machine breakdown, rush job and changes in job priority, labour absence, and late arrival of material.

Impact of these events on system performance are often difficult to estimate. It is because manufacturing operations are characterised by the existence of numerous dependent events and interactions between resources and products. The dependent events come from the constraints imposed on the manufacturing process, such as operation sequence of an order. The interactions would normally be used to indicate the effect that dependent events have upon one another [Umble & Srikanth 1990]. Due to the existence of dependent events and the interaction between resources and products, the impact of a disruption event could propagate in the system.

Of the disruptive events, machine breakdown is one of most commonly occurring events and has great impact on system performance. Therefore, the control strategy to deal with machine breakdown is one of the most important elements of real time shop floor control strategies.

4.2.2 Important issues for determining control strategies for dealing with machine breakdown

Although much effort is usually made by companies to prevent machine breakdowns, e.g. regular preventive maintenance, machine breakdown is still a commonly encountered disruption on the shop floor.

Machine breakdown is a difficult problem to deal with. It is not only because such an event could happen randomly, but also because of the ways by which it would affect system performance. A machine breakdown can affect system performance both during the period in which the machine is broken down and the period after the machine has recovered. During the period in which a machine is broken down, the event has the following major effects on the system:

- a) the workcentre is losing available capacity, and it may become congested or even a bottleneck.
- b) Jobs will be delayed or blocked at the workcentre. As a result, its downstream machines may experience a period of loading valley, and low utilisation. Some of the downstream machines may be forced to shut down due to lack of material and, hence lose capacity.
- c) The jobs queuing or blocked at the workcentre could become critical.

After the machine has recovered,

- a) the jobs on the workcentre may become late or critical, and their uncompleted operations may, therefore, have to be given high priority to access required resources for meeting due date requirements.
- b) As a result, jobs may be congested at workcentres downstream from these recovered machines. In other words, these downstream workcentres may experience a peak loading period and job congestion.

It is obvious that machine breakdown could affect a plant's ability to maintain the required production flow in a smooth and timely manner. More seriously, it could cause 'moving' bottlenecks. Therefore, for effective shop floor control, there is a need to include both machine and other resource related information in the dispatching process. A dispatching procedure which uses both order related and resource related information can be seen as having multiperspectives.

4.3 Discussion on approaches of multiperspective dispatching

4.3.1 Review of approaches related to multiperspective dispatching

One approach to address this issue is to dynamically switch between order-based and resource-based perspectives. [Smith & Ow 1990] proposed this approach in multiperspective scheduling, and OPIS is able to dynamically switch between order-based and resource-based perspectives in schedule generation and repair.

For dispatching, [Wu & Wysk 1989] developed a multipass expert control system (MPECS) by which a formal dispatching rule can be selected from a set of priority rules based on real-time information. In the research, two machine related rules were included:

WINQ Select the job for which the next operation is at a workcentre with the least work load.

NINQ Select the job for which successor workcentre has the shortest queue.

COVERT (cost over time rule) and SPT were also included, respectively for taking into account both remaining work and cost, and for maximising throughput.

There are some other approaches proposed for multiperspective dispatching. For example, [Raman & Talbot 1993] developed a heuristic approach, which established relative job priorities using operation due dates (ODD), but taking into account the impact on other jobs in the system, and also used the relative workload of a given machine to determine criticality of orders. Many approaches suggested for control of FMS use both order and machine information, [Ben-Arieh et al 1989] [Dutta 1990], but, as noticed by [Gupta et al 1989], maximising machine/system utilisation were given importance by researchers although shop floor personnel often see meeting due dates as more important.

[Schonberger 1979] suggested a strategy called Clearest-road-ahead for dispatching jobs in a typical discrete part production system. By his definition, Clearest-road-ahead is “the routing that passes through under-loaded work centres,” and “cloudiest road ahead is the routing that passes through overloaded work centres”, and the approach attempted to dampen workload unevenness in the work centres by using work centre utilisation in priority determination. [Schonberger 1979] proposed this approach to be used as a formal, computer-aided method for dispatching. He suggested that orders with clearest-road-ahead should be given high-priority in dispatching, and in contrast, orders with cloudiest-road-ahead should receive low-priority in dispatching. He attempted to provide this work centre underloaded and overloaded data in the daily dispatch list to alert the shop floor personnel to the need for either speeding-up (pulling-ahead) of orders moving next to underloaded work centres, or delaying (dropping-back) of orders moving next to overloaded work centres and, thus, providing a smoother workflow. [Schonberger 1979]

In shop floor control practices, as noticed by [Melnik 1988], shop floor personnel often take into account machine related problems in making dispatching decisions, such as disruptive events and bottlenecks which constrain overall capacity. They are sensitive to the existence of these bottlenecks and they see the presence of

capacity problems, e.g. bottlenecks, as information which should be used in modifying the resulting priorities of jobs.

In DRC (Dual Resource Constrained) production environments, use of information about the anticipated workload at related work centres when making dispatching or labour assignment decisions was also described by [Trelevent 1989] as a “potentially fruitful avenue for future research”.

4.3.2 Important issues on management of bottlenecks

In general, bottlenecks can be seen as constraining resources which determine the effective capacity of the system and the level of system output. But there is a lack of agreement in the literature about what a bottleneck is.

4.3.3 Definition of bottlenecks

Bottleneck is often defined from the following dimensions:

- 1) resource utilisation,
- 2) length of job queue waiting for a resource,
- 3) job arrival rate to a resource against the rate they leave the resource.

[Prather 1983] defined bottlenecks as “any work centres which have capacity utilisation above 90 percent or the longest queue.” [Wallace 1980] defined bottlenecks as “facility, function, department, etc, that impedes production - for example, a machine or work centre where jobs arrive at a faster rate than they leave”.

But, as pointed out by [Melnik 1988], these definitions only identify bottlenecks after the fact, and do not provide information about the cause of the bottlenecks,

which could be due to short term problems on the shop floor or the result of long term capacity limitations. In other words, some critical characteristics of bottlenecks should be considered and described for designing and implementing control strategies.

4.3.4 Critical characteristics of bottlenecks

[Melnik 1988] gave following examples of bottleneck characteristics:

- Nature of the bottlenecks: Bottlenecks can be described as either stationary, (i.e. associated with only a particular resource all the time) or moving (i.e. at a given point in time, a resource appears to be a bottleneck, and at a later time, that same resource can have excess capacity).
- Number of bottlenecks: single (only one bottleneck) or multiple bottlenecks,
- Location of bottlenecks, Bottlenecks can occupy one of three positions: front, e.g. gateway workcentres; exit, such as a constrained workcentre which is at the end of an order's routing, and random, which means that bottlenecks can occur anywhere on the shop floor.
- Prevalence of the bottleneck, A characteristic of a bottleneck which describes the extent to which jobs flow through it. It could be the case that all jobs use the bottleneck, or only a small percentage of the jobs require it. When prevalence is less than 100 percent, it is important, as suggested by OPT philosophy, to distinguish between jobs which flow through the bottleneck and those which do not.

Characteristics of stationary bottlenecks can be recognised and taken into account in the planning phase by some approaches, e.g. by OPT or the approach proposed by [Ow 1985]. Characteristics of moving bottlenecks, however, are difficult to be described before production since these bottlenecks occur during production and

could move around from one resource to another, wandering among the resources in the system. The problem of moving bottlenecks can only be dealt with by real-time control mechanisms whenever the phenomena has been detected on the shop floor.

As observed by [Melnik 1988], in effective shop floor control systems, shop floor personnel often use resource related information, and their knowledge and strategies to cope with contingency and to improve effectiveness of the dispatching mechanism. Particularly, he noticed [Melnik 1988] that shop floor personnel are constantly monitoring and using information on both upstream and downstream workcentres, whenever possible, to modify operation priorities (e.g. giving higher priority to orders going to relatively underutilised work centres and lower priority to orders going to bottleneck resources.) A key issue is how to coordinate order and resource related perspectives. A dispatching decision can be made based on current shop status, e.g. information about urgency of immediate operations and loading levels of workcentres, but such a decision is 'shortsighted'. A look-ahead approach is often used for extending the view of the dispatching mechanism.

4.3.5 Review of approaches for look-ahead

Look-ahead implies evaluation of the consequence of a decision by projecting the shop status. The commonly used basic method behind a look-ahead function is the use of deterministic simulation as a short term predictive tool for evaluation of alternative dispatching options. There are two major problems in using a look-ahead function, efficiency and accuracy.

Efficiency refers to the ability of a look-ahead mechanism to provide predictive information within a permitted time interval; and accuracy refers to reliability of the predictive information.

Off-line, time-consuming simulation is sometimes suggested for shop floor control,

but the opportunity for providing valid and timely answers to ‘what-if’ questions in the dynamic manufacturing environment are limited.

The accuracy of predictive information is influenced by uncertainty levels in the manufacturing environment. In much research on scheduling, e.g. [Gere 1966], it is assumed that there is no uncertainty, and the predictive information obtained by look-ahead mechanism will be reliable. In some others, especially research carried out for studying a specific system, some parameters of unexpected events were assumed to be known, e.g. [Ben-Arieh et al 1989] used fixed machine repair times. In a dynamic manufacturing environment with random disruptive events where all the parameters are unknown, (e.g. time, place and repair time of a machine breakdown), the reliability of predictive information is decreased, and the usefulness of this approach may depend upon other factors, the most important of which is a look-ahead horizon.

[Ben-Arieh et al 1989] used three levels of look-ahead horizons (all future operations, next two operations, and only one operation) to test its sensitivity against three frequency of machine breakdown events, and the results (throughput of assembled parts) showed full look-ahead under steady conditions (no breakdown) to yield the best outcome. Under a moderate failure rate, the full look-ahead option is still better to use than the other options. For a higher failure rate, however, the medium look-ahead option is shown to be the best. He concluded “It is evidence that in a noisy environment a long plan-ahead knowledgeable heuristic does not function effectively because of the frequent changes in policy driven by external unpredicted failures.” And therefore, he suggested the use of different look-ahead horizons for different frequencies of machine breakdown events. But, in many manufacturing environments, as discussed above, any parameter of a machine breakdown event, including the repair time, can not be estimated with any level of confidence.

Since [Ben-Arieh et al 1989] used fixed repair times in his experiments, total ma-

chine downtime in a given time span within the plan was virtually proportional to the frequency of machine breakdowns. Therefore, when he suggested the use of different look-ahead horizons for different frequency of machine breakdown events, he was suggesting that both the level of total machine downtime as well as the time interval between machine breakdowns be taken into account.

In general, it is not realistic to look ahead and anticipate every conceivable conflict to seek optimal solution in the dispatching process, but this function, as suggested by [Gere 1966], can be used to just look ahead a short way at a modest cost. In dispatching, this function can be used to look ahead and anticipate conflicts between a scheduled operation and other jobs, and check to see the effect of this action on other jobs. [Gere 1966] used a look ahead heuristic to check if there is a late or nearly late job due to reach the dispatching machine before a selected operation can be completed, and used the result for deciding if there is need to select another operation instead. [Zeestraten 1990] proposed a look ahead dispatching procedure for minimising makespan in a job shop with routing flexibility.

A look-ahead function plays a key role in coordination of order and resource perspectives, and forms a basis for integration of objectives of meeting job due dates and capacity management. However, due to the unpredictability of future events, the extent of the look-ahead may need to be limited.

4.4 Conclusions on dispatching strategy

As discussed in 4.1, a simple rule based dispatching strategy is unable to effectively maintain shop floor stability, and it is essential to use information beyond the immediate jobs, i.e. dispatching mechanism needs broader and strategic view of shop status. Two important issues have been discussed in the last section. One is multiple perspective based approaches, i.e. strategies using both order related

and resource related information, e.g. OPIS and Clearest-road-ahead strategy; the other is use of information on future shop status, such as that in the research by [Ben-Arieh et al 1989].

A multiple perspective view of the problem of dispatching emphasises the need to include resource related information in the dispatching process since in a dynamic manufacturing environment resource availability often becomes one of the major constraints in maintaining shop floor stability. In single resource environments, information on the next workcentre is often used, e.g. [Wu & Wysk 1989] used NINQ and WINQ rule. In a DRC production environment, however, information on the dispatching workcentres needs to be included for queue selection in addition to information on the next workcentres. A multiple perspective view also emphasises the importance of detecting and identifying dominant constraints, and appropriate coordination of multiple perspectives in the dispatching process.

In industrial environments, resource related information, especially information on upstream and downstream workcentres, is frequently used by shop floor personnel for improving the effectiveness of shop floor control, as observed by some researchers [Melnik 1988]. In a discrete manufacturing environment, there are usually a number of work centres and each order usually has a number of operations. In such a manufacturing environment, the sequence in which jobs are to be processed at the succeeding work centre could be critical for determining the sequence in which they should get completed at the current work centre. There is also a need for consideration of jobs on upstream machines since the sequence in which resources are assigned in one workcentre could affect opportunity of the jobs on upstream machines to access required resources in a timely manner.

For coordination of multiple perspectives in dispatching, it is important to make decisions based on both current and future state of the shop; in other words, a dispatching decision should not only be good for current shop status but also should have a positive effect on future system performance.

Assigning resources to a job can have both immediate effects, e.g. starting up setting or processing of the job for meeting the job's due date, delaying other jobs which are currently competing for the same type of resources, etc, and future effects such as causing jobs which are currently on upstream work centres to wait, and the selected job is going to be blocked or delayed at succeeding workcentre after it has completed its operation on the current machine.

Thus, dispatching needs to be based not only on current shop status, but also on knowledge about future effect of a dispatching action. This kind of knowledge usually can be obtained in two ways, intuition and Look-ahead function.

Shop floor personnel often use intuition to fill in the blanks about what is happening on the floor, what can happen, and what will happen, and then make decisions mainly from their experience, intuitions or guesses. For instance, when a workcentre is broken down and the jobs blocked will be waved downstream after the workcentre has been repaired, intuition suggests that its downstream workcentres may become congested sooner or later, and dispatching for these downstream workcentres should then take this possible effect into account.

But, due to complexity of the shop floor situations and limited time, intuition based prediction about the effect of a dispatching action is, in many cases, unable to provide quantitative information which can be used in the dispatching process. Therefore, computerised look-ahead approach is often used. As discussed in the last section, due to time constraints and dynamic nature of the environment, it may be more appropriate for a dispatching mechanism to just look ahead a short distance.

To implement a dispatching strategy based on the approaches described above, a key issue is how to establish the relations between the goal, i.e. maintaining shop floor stability, and the available information, including current and future

shop status for job selection. In situations where violation of objectives of control, e.g. missing due dates, can be detected, the objective related goal functions, e.g. total tardiness, could be used. In other situations, no violation of objectives may be detected (it does not mean that selection of a particular job would not cause such violation, it is more possible that it is because the look-ahead facility in the dispatching mechanism is unable to anticipate the conflicts among all the remaining operations); in these cases, the planned operation starting and completion time in the predictive schedule will provide useful intermediate objectives for job selection. In essence, (since objectives are constraints themselves), a core function of such a control strategy is to identify the dominant constraints at the time of dispatching.

To identify the dominant constraints, all the constraints which may have significant effects on the outcome of the control process, including order related and resource related information, need to be classified, described and quantified (if necessary). Order related constraints can be measured by the urgency of the orders in terms of meeting their due dates, while resource related constraints by status (e.g. broken down, busy, or idle), length of waiting job queues, remaining work level, etc. A control strategy can be seen as a method to define, describe and use these constraints to reach a decision.

Since analytical approaches are unable to describe such a complex problem, heuristic approaches are usually used, e.g. all the approaches reviewed in last section are heuristic approaches. Knowledge based approaches are particularly favoured by many researchers and practitioners due to their ability to facilitate manipulation and coordination of heuristic knowledge and quantitative data.

It is believed that the strategy of appropriate coordination of multiple perspectives in the dispatching process could lead to an effective way to maintain shop floor stability.

Chapter 5

A proposed shop floor control strategy

5.1 Outline of the proposed shop floor control strategy

In this research, a control strategy is proposed for improving performance of the shop floor control in a job shop or discrete batch production environment with dual resources, machine and labours, under disruption of machine breakdown.

This control strategy can be described by its paradigm.

5.1.1 Paradigm of the proposed control strategy

The paradigm of the proposed control strategy can be described by the types of the control actions used and its dimensions.

The proposed control strategy uses the two most commonly used real-time control actions, job reallocation and overtime. As discussed in chapter 4, it is the belief that the key to improve effectiveness of real-time shop floor control un-

der the disruption of machine breakdown is to smooth flow of work among the workcentres.

The dimensions of the control strategy, as discussed in chapter 4, can be described from an order perspective and a resource perspective.

From an order perspective, three groups of jobs could be related to a dispatching action (job selection):

- candidate jobs for which machine and labour requirement can be met at the time of dispatching,
- the jobs which are going to access the same types of machine and/or labour as a selected job and, hence going to be delayed,
- and the jobs which are using or going to acquire the same types of machine or labour as used by succeeding operations of a selected job.

Jobs in different groups can be affected by a dispatching action in different ways. The candidate jobs can either be assigned to resources or be delayed, i.e. their prospect of meeting due dates would be directly affected; the jobs in the second group are going to be affected, i.e. delayed, due to a lack of machine or labour as a result of a current dispatching action; and the jobs in the last group can affect or be affected by the chance and/or timing of the remaining operations of the candidate jobs to access required resources in the future.

Considering the dynamic nature of the manufacturing environment, and the very short time horizon of the real time control actions, the proposed control strategy only uses information which are reliable and are close to the decision point. As shown in **Figure 5.1**, the proposed control strategy uses information about the jobs on direct upstream machines of the dispatching workcentres, and the jobs that are engaged on (in queue or on machine) or going to arrive at the succeeding (direct downstream) workcentres at the time of the dispatching.

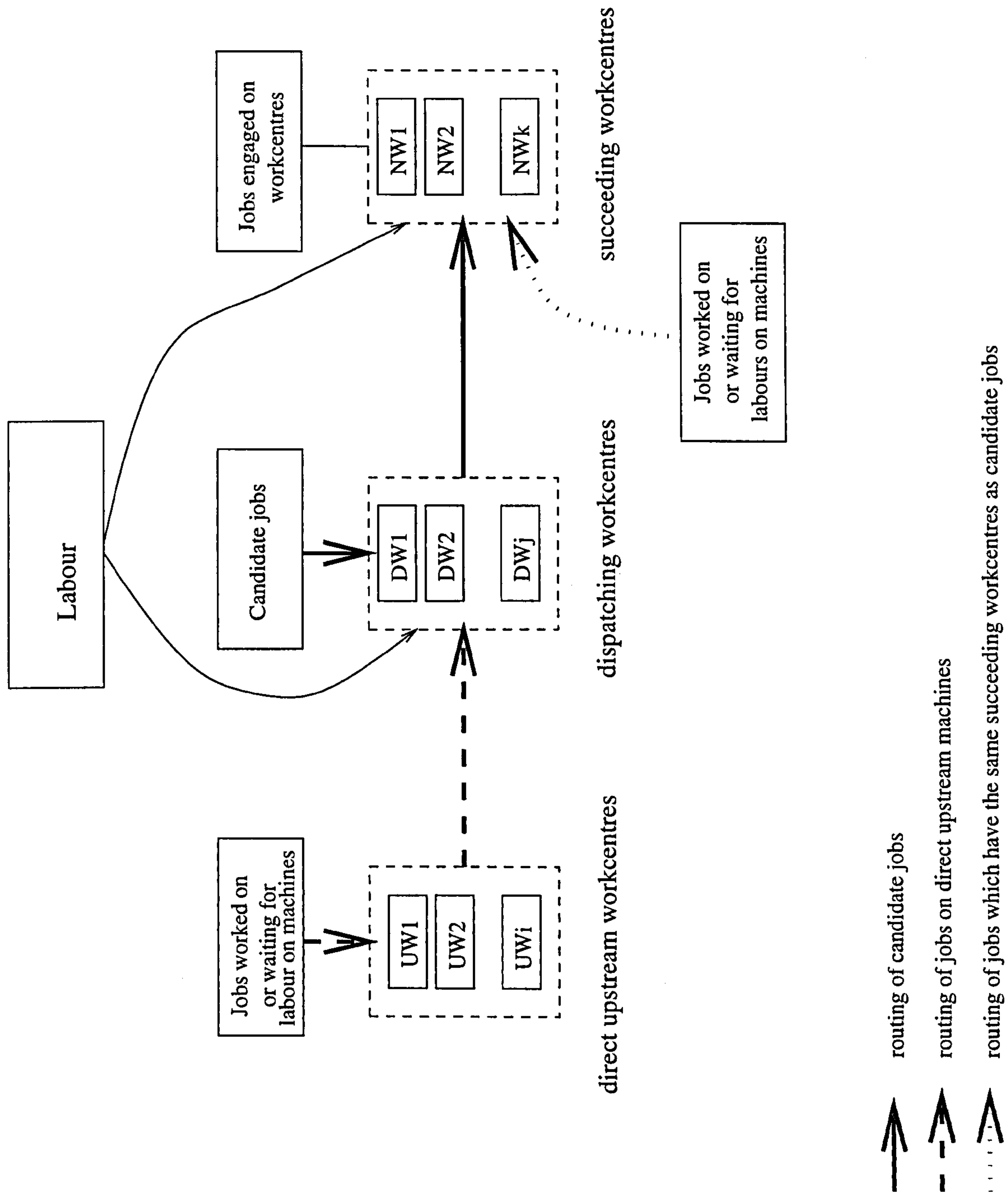


Figure 5.1: Jobs and workcentres for which information have been included in the proposed control strategy

From a resource perspective, the availability of machines and labours is seen as constraints by the control mechanism, e.g. a job on a broken down machine can not be selected. Control is seen as a means of levelling loads among the resources. In a dual-resource environment, load smoothing can be done between labour or between workcentres. When selecting a job on a workcentre, utilisation of other dispatching workcentres, if there is another one, may be affected, and utilisation of workcentres on the routing of the job's remaining operations would be affected in the future. From this view, job selection can be seen as a way of levelling the work loads among the dispatching workcentres as well as the downstream workcentres.

The proposed control strategy also suggests a way to coordinate the capacity management related factors with order related factors for coping with disruptions due to machine breakdowns.

5.2 Approaches for coordination of order related factors

Meeting due dates is often considered as the most important objective for production planning and control. In the proposed control strategy, three major factors - candidate job due date priority, consequence of a job selection, and status of the next workcentre have been taken into account from the perspective of meeting due date requirements.

5.2.1 Approaches for description of candidate jobs' due date priority

Two methods have been used for describing candidate jobs' due date priorities,

One is priority groups, the other is priority rankings derived from a priority rule.

Candidate jobs can be classified into following four priority groups according to the level of urgency for meeting its due date, and whether the job is lagging behind the timing plan contained in the original schedule:

- 1) late jobs are jobs for which the current operation has already missed operation due date,
- 2) critical jobs are jobs having low slack time. In this research, a critical job is regarded as a job which has slack time less than 10 percent of its remaining lead time,
- 3) due start jobs include two group of jobs. One are the jobs for which the current operation is due to be assigned to resources according to the predictive schedule, but is neither late nor critical; the other are jobs for which the current operation is not due to be assigned to resources according to the predictive schedule, but has a priority rate higher than a candidate job in the first group,
- 4) non due start jobs are those for which the current operations are not due to be assigned resources according to the predictive schedule, and their priorities are not higher than that of any due start job.

The purpose of using priority group is for identifying the level of urgency and stage of progress of the candidate jobs. By using job due dates, late jobs and critical jobs can be detected for the control system to give these jobs priority and let them catch up or prevent them from becoming late jobs. By the predictive schedule, job selections can follow the schedule when there is no disruption or the plan is still good to be followed; identification of due start jobs and non due start jobs provides some indicators as basis for schedule repair.

In many industrial situations, the planning system usually generates feasible plans

by a finite scheduling mechanism, i.e. the orders released by the planning system to the shop floor do not require access to more shop resources than are currently available if there is no occurrence of disruption events or random fluctuations. Such a realistic, (or good, or best) predictive schedule can provide information for closely monitoring and controlling jobs' progress on the shop floor. Without disruption, start times and time segments for each operation in such a predictive schedule can be used as intermediate objectives for conducting dispatching activity. After the occurrence of a disruption event, the predictive schedule may need to be adjusted by changing the sequence of jobs, overtime, etc. As discussed in chapter 4, the impact of a disruption event might be propagated to the whole system. But, in many cases, resequencing of a job (or some jobs) could make the schedule match up to the original plan, or make the schedule be 'good' enough as basis for control.

The jobs in the same group can be seen as 'critical equivalent' jobs with a similar due date priority band and stage of progress, and the ultimate jobs' priorities in the same group would be determined by other factors.

Priority rankings which are derived from ODD rule are also used for determination of ultimate priorities of the jobs in a same priority group, but it is used after taking into account the consequence of a job selection.

5.2.2 Consequences of dispatching actions

Allocation of machine and labour to a job could result in new constraints on the availability of these resources, and these constraints could affect the chances of other jobs' access to the same types of machine or labour. Including such information in dispatching will allow the control system to have a broader view of the objective of meeting job due dates, not only from the view of the candidate jobs but also of the other jobs in the system.

These effects can also be measured according to due date objectives and the original schedule:

- a) if and how many jobs (operations) are going to miss their due dates, i.e. number of increased late jobs,
- b) increased total tardiness,
- c) if and how many critical jobs (operations) are (going to be) delayed
- d) if and how many jobs (operations) are going to miss their due start time predetermined in the predictive schedule,

By taking into account these factors, the dispatching could be carried out towards the objective of meeting due date requirement for the candidate jobs as well as for other jobs in the system. By this view, four levels of shop floor situations can be identified at the time of dispatching:

- 1) when selection of any job leads to violation of one or more due dates, i.e. causes added total tardiness (late job situation).
- 2) when the least adverse effects of a job selection is to cause a critical job to be delayed (critical job situation).
- 3) when the least adverse effect of a job selection is to delay a due start job according to the original plan (due start job situation).
- 4) when there is a job which can be selected without causing any adverse effects (no adverse effect situation).

In addition to these factors, as discussed in chapter 4, the perspective of meeting the order due date should also be viewed from the chance and timing of the job to access required machine and labour at its next stage of operation after completion of the current one.

5.2.3 Approach of 'look-road-ahead'

As discussed in chapter 4, since most of the orders have to go through more than one stage of production, the prospect of a selected job to meet its due date at the time of dispatching is not only determined by the decision made for assigning resource to the job's current operation, i.e. allow the job to go through its current stage of operation as soon as possible, but is also influenced by its chances to access required resources at its next stage of operation after completion of the current one. In other words, the dispatching system should have a 'look-road-ahead' mechanism.

There are two factors to consider in a 'look-road-ahead' mechanism - the condition of the next workcentre and the urgency of the job which will go through it. For example, a late job should go down the 'road' even if it is congested. In this research, the following information concerning the prospect of meeting the selected job's due date at its next workcentre is included for decision making in the dispatching process:

- a) Status of the workcentre, whether it is broken down or not,
- b) Predicted waiting time of the selected job at the workcentre.

If the workcentre is broken down, the selected job may have to wait at the successor workcentre after completion of the current operation, especially when there are some unfinished operations at the broken down workcentre.

The waiting time of a job at its next workcentre is determined by several factors when the next workcentre is not broken down. The major one is the due date priority of the job at the next stage of production.

A look-ahead mechanism is used to provide information about the consequences of a job selection as well as the job's waiting time at its next workcentre.

5.2.4 Function of the look-ahead mechanism

For efficiency of the look-ahead facility, and obtaining relatively reliable information of future shop status, this mechanism projects partial shop floor status and only looks ahead a short distance.

The function of the look-ahead facility is to anticipate conflicts between jobs on the dispatching workcentre(s), direct upstream and downstream workcentres, and the jobs which are going to compete for the same types of machine or labour with the selected job.

The look-ahead facility has two phases of operation: generation and evaluation, which are respectively for projecting shop status, and evaluating and translating the future shop status into information in a format usable for decision making.

In the generation phase, the look-ahead facility would establish the job sequence on the dispatching workcentre(s) and the workcentres on which there is an operation which is going to be delayed by the job selection due to the constraints of labour availability imposed by that job selection.

Operation sequence on these workcentres will be generated by projecting from the current shop status. For each workcentre, the jobs which are currently engaged on a machine at the workcentre, or are going to be operated on the workcentre, will be taken into account. They include:

- a) Jobs which are currently being set up, processed, or waiting for labour on a machine at the workcentre,
- b) Jobs which are currently queuing in front of the workcentre,
- c) Jobs which are currently engaged on a direct upstream machine of the workcentre,

Jobs which currently are on broken down machines are not considered by the look-ahead facility.

Sequence of these operations are determined by:

- a) Earliest material arrival time,
- b) Earliest machine and labour available time,
- c) Due date priority ranking by ODD rule.

By taking into account these factors, the sequence and predicted start times of the operations on these workcentres can be generated. And then, in the second phase, the predicted start times will be compared with their due dates, as well as the planned start times in the original schedule. Thus, some essential aspects of current and future time and capacity constraints of each candidate job could be identified, and used as basis for determining the control actions.

The look-ahead mechanism virtually looks ahead two operations, current operation and next operation at succeeding workcentres for each candidate job.

By means of the look-ahead mechanism, the control system can have a picture of future shop status from meeting job due date point of view for each job selection, and make decisions based on this information.

5.2.5 Approaches for coordination of the due date related factors

Coordination of the due date related factors in this research tries to take into account three major factors: due date criticality of candidate jobs, consequences of job selection and status of succeeding workcentres.

By the proposed control strategy, a subset of the candidate jobs are first selected according to two of these factors: the jobs' criticality level, described by the priority group they belong to, and consequences of their selections, measured by added tardiness, number of affected critical jobs, etc.

As shown in **Figure 5.2**, the process of this selection starts from highest priority group, and the jobs having the least adverse effects in the group are more likely to be chosen. But, jobs in a lower priority group are selected when the jobs in higher priority groups all lead to worse detected consequence, such as:

- when selection of any job in a higher priority group causes added total tardiness, but selecting a job in a lower priority group would not;
- when selection of any job in a higher priority group causes a critical job to be delayed, but a job in a lower priority group would not.

This could happen when a production process has deviated from its original schedule due to machine breakdown. For example, when there is a job which was interrupted in the middle of an operation by a machine breakdown and is late or critical after the machine has recovered, the main concern in job selection on its succeeding workcentre will be to avoid or reduce an adverse effect to the job on the direct upstream machine if current candidate jobs are less urgent, e.g. due start jobs and non due start jobs. In this situation, a non due start job can be selected for this purpose even if there is a due start candidate job to ensure that the critical job on the upstream machine has a quicker passage.

As a result, the jobs first selected may not be in the same priority group. This process can be seen as necessary for the purpose of identifying different situations (late job related situations, critical job related situations, due start job related situations, and no adverse effects situations), so that the control system could use

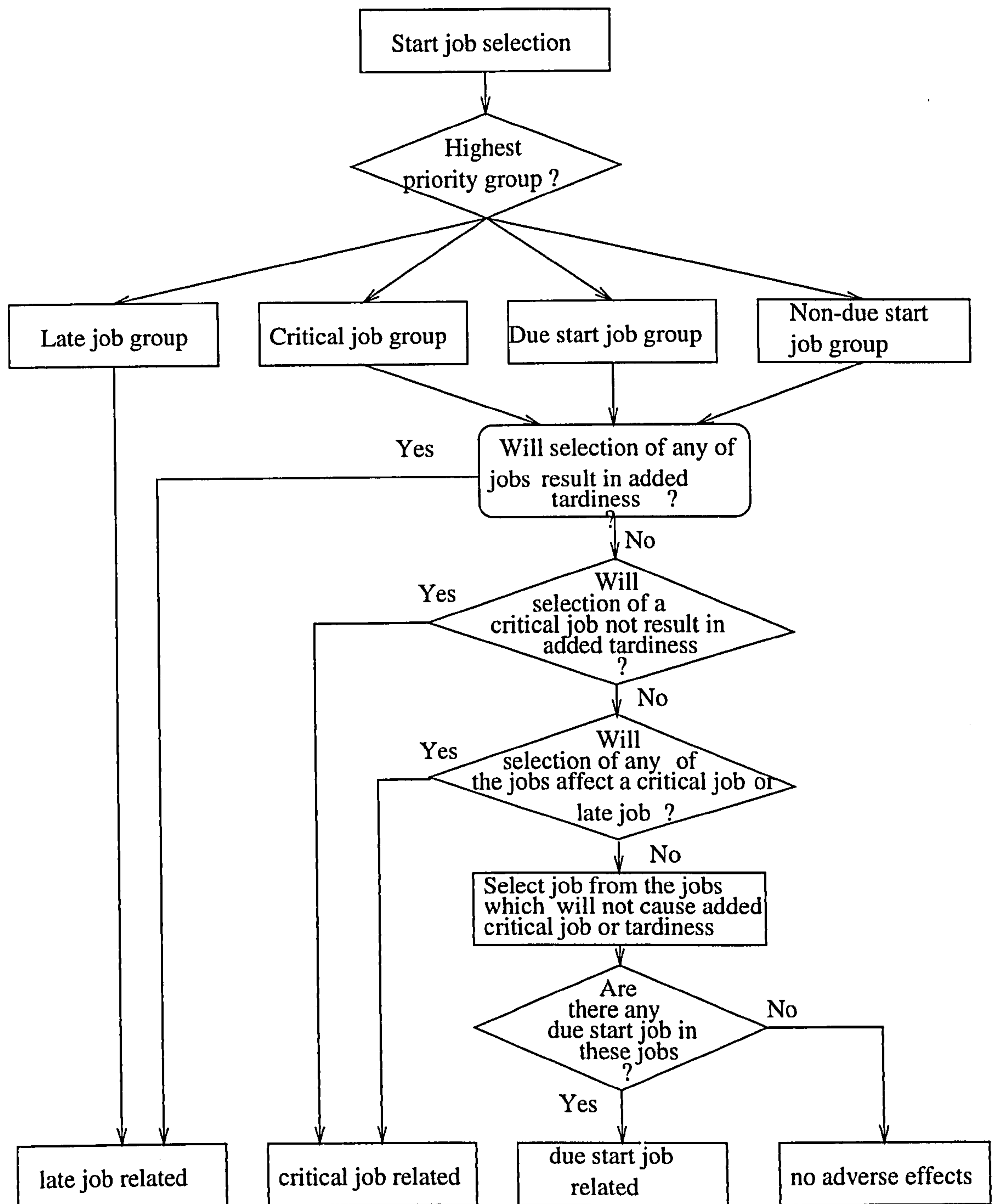


Figure 5.2: The control process for identifying levels of effect of job selection

different policy to conduct job selection, and to include other factors, e.g. status of succeeding workcentres.

The following approaches are suggested by this research for coordination of the due date related factors in different situations:

tardiness (TA) - an approach by which tardiness caused from dispatching actions are used as a measurement for job selection. This approach is used in late job related situations, where there is a late candidate job, or any job selection will cause added tardiness. The measured tardiness includes both the tardiness resulting as a consequence of a job selection, and the possible tardiness at the next workcentre due to unavoidable queuing time. In these situations, a job which causes minimum increased total tardiness will be selected.

critical job (CJ) - an approach which is used in the critical job related situations, where there is a candidate job which is critical, or selection of any job will cause one or more jobs to become critical or will delay critical jobs. In these cases, job selection will be according to the number of critical jobs affected, i.e. a job which will cause the least number of critical jobs to be affected will be selected.

due start job policies (DSJ) If there is only one due start job, it will be selected. If there is more than one due start job competing for the same type of machine or labour, job selection will be determined by other factors, e.g. planned due start time, due date priority derived from ODD rule, and information from 'look-road-ahead' mechanism. When all the candidate jobs are non due start jobs, and selection of some or all of these jobs would cause neither critical job nor late job, job selection could be based on the measures about if and how many operations will miss due start time as consequences of that job selection.

look-road-ahead (LRA) a policy for selecting jobs according to waiting times at the next workcentres after completion at the current workcentres. When

selecting a late job, such waiting times are included in total tardiness. In due start job related situations, if there is more than one due start job competing for the same type of machine or labour, two methods are used; both are based on jobs' due date priority ranking (derived by the ODD rule) and predicted waiting time at the next workcentre. One can be called the 'short queuing time' rule by which a due start job with high due date priority ranking and short predicted waiting time at its next workcentre will be selected. The other is called 'queuing time compression', by which a due start job that has short predicted waiting time at its next workcentre, and selection of which will not delay start times of other candidate jobs with higher priority rankings that are on the same workcentre as itself, will be selected; in the other words, selecting the job only leads to compression of inevitable queuing time of these higher priority ranking due start jobs at their next workcentres. In this study, short waiting time is measured by mean waiting time of a queue which is calculated according to the current shop status. If a job's next workcentre is broken down, or the job's predicted waiting time at the next workcentre is relatively long, other jobs could be assigned to take advantage of its unavoidable queuing time at the next workcentre.

As mentioned before, in many job shop and discrete batch manufacturing systems, there is some robustness in production plans. A large portion of a product's manufacturing lead time is spent waiting in queues at various work stations in the system. For resources, nearly every schedule has capacity allowance, or a time buffer, to protect the throughput of the system from disruptions that continually occur in the manufacturing environment. Due to existing robustness in the schedules, there are some situations, such as 'due start job' and 'no adverse effects' situations, where there is no imminent risk of violation of due date objectives and there is some flexibility for the dispatching mechanism to compress jobs' queuing time as part of its capacity management functions; it helps to better distribute the slacks available for various jobs/resources.

The general idea behind these approaches is:

- In the situations where tardiness increase is unavoidable, control should be based on the objective of reducing total tardiness.
- In the situations where there is a risk of violation of job due dates, it should be prevented,
- In the situations where there is no risk of violation of job due dates, but the candidate jobs are due to be assigned resources, choose the one by taking into account condition of succeeding workcentres.

When more than one job has same highest priority ranking by means of the above factors, some other factors would need to be used to make decision. If there is a due start job, critical job or late job, ODD rule is usually used. If they are all non-due start jobs, load smoothing related factor are to be considered; job selection could be conducted for purpose of smoothing work loads, i.e. real-time capacity management.

5.3 Approach of real-time capacity management

As discussed in chapter 4, resource capacity and availability often constrain work flow on the shop floor and one of the major tasks of a shop floor control system is to detect such problems and prevent and/or reduce their impact on system performance. In a manufacturing environment with disruption of machine breakdown, a workcentre's capacity and timely availability often become the major constraints and create bottlenecks.

5.3.1 Definition of bottlenecks

In this research, constraints on workcentre capacity and availability is detected

by its current available capacity as well as the queue length.

Bottlenecks are defined as the workcentres for which remaining work is over 90 percentage of available capacity in the schedule horizon. The bottlenecks can further be categorized as overloaded or non-overloaded for the control systems to identify the major constraints.

Since the length of a job queue could also have significant influences on timely flow of work, long queue and potential long queue workcentres are also identified as constrained workcentres in this research.

Definition of a long queue and potential long queue workcentre is based on a system mean value, current average number of jobs at each workcentre, i.e. mean queue length.

A workcentre is defined as a long queue workcentre if the number of jobs queuing at a workcentre is twice the mean queue length for the system; a workcentre is defined as potential long queue workcentre if queue length of the workcentre is not equal to or greater than twice the mean queue length, but the sum of the number of jobs queuing and number of jobs which are going to arrive at the workcentre is three times that figure. To define a potential long queue workcentre, only the jobs which have already been engaged on direct upstream machines of the workcentre, i.e. have been loaded on machines and in the middle of setting up or processing, are taken into account; jobs on broken down machines are not included.

This definition of bottlenecks provides the basis for the shop floor control systems to identify the workcentres which constrain the flow of work, and to level loadings on different workcentres in dispatching.

5.3.2 Approaches for capacity management in dispatching

Dispatching can be seen as a means of smoothing loading levels among different workcentres. A decision to allocate resources to a particular job has a direct effect on the queue of its current workcentre and, in due course, will also affect the queue of the workcentre for its succeeding operation.

In many situations during production, labour could be a limited resource. In these cases, decisions on assigning labour to different workcentres (different job queue) can be a way of levelling work loads among the dispatching workcentres. At the time of dispatching, all the workcentres in the system can be classified at three levels:

- 1 overloaded bottlenecks,
- 2 non-overloaded bottlenecks, long queue or potential long queue workcentres,
- 3 others.

Since overloaded bottlenecks directly affect due date objectives and production throughput, overloaded bottlenecks are given highest priorities. The workcentres in level 2 represent those which could constrain the flow of work, but such constraints may not have imminent effects on system performance and therefore, are given lower priority. Other workcentres are given the lowest priorities as they are not seen to be constraining the flow of work.

The following control policies are used for selection of job queue in this research:

Keeping bottleneck busy (KBB) This policy is for bottleneck management, and avoiding congestion, especially to overloaded workcentres. By this policy, jobs on bottleneck workcentres and long queue and potential long queue workcentres will be given a high priority in dispatching.

Downstream machine (DM) a policy for preventing the problem of congestion after broken down machine(s) have recovered and jobs begin waving downstream. By this policy, jobs downstream of a broken down workcentre are given high priority to obtain labour.

Dispatching can also be used for smoothing work load among its downstream workcentres, especially the direct succeeding workcentres. By the proposed control strategy, information about status and loading of next workcentres are used for the purpose of capacity management. The major policy for next workcentre selection is the idle machine (IM) rule proposed by [Buzacott 1982] for job release control. It selects a job for which the next workcentre is idle or going to be idle, specially when such a workcentre is a bottleneck.

More detailed information can be used in dispatching for smoothing loads among the downstream workcentres, such as:

status If a workcentre is broken down, i.e. all the machines in the workcentre are broken down, or has at least one idle machine, or has at least one busy machine,

bottleneck if it is a bottleneck, long queue or potential long queue workcentre,

job queue number of jobs waiting in the queue,

arriving jobs number of jobs engaged on its direct upstream machines, (being set up, processed, or waiting for labour),

remaining work (RW) total unfinished work on the workcentre.

predicted queuing time predicted queuing time of a candidate job at the workcentre.

The major purpose of load smoothing is for preventing and/or reducing the risk

of capacity problems caused by machine breakdown, e.g. machine starved of work, congestion, etc. These problems, as discussed in chapter 4, will sooner or later prevent timely flow of work, and result in adverse effects on objectives of maintaining shop floor stability. It is believed that including a real time capacity management function in the dispatching procedure would more effectively smooth loading among the workcentres and result in global and positive effects for maintaining shop floor stability.

5.3.3 Overtime based capacity control

Overtime is another real-time capacity management means used for compensation of capacity losses in a production process. For maintaining shop floor stability, the goal of overtime is to recover as much as possible late jobs and tardiness with as little as possible overtime costs.

There are generally two overtime policies, reactive and proactive [Scudder 1987]. The reactive policy is to use overtime when there is a late job in the system, while proactive is for compensating for low capacity of resources or when there is risk of inadequate capacity to cover expected demand. Both of the policies have been used in this research. The reactive policy is obviously required for recovering late job and/or reducing total tardiness. The proactive is for compensation of overloaded workcentre and preventing some job becoming late due to lack of machine capacity in the future. When overtime is possible, i.e. at end of a shift and/or end of day, conditions to use overtime can be described as follows:

- When there is a late operation which can be worked on (set up or processed) during overtime, or
- When continuing with an operation in the overtime will reduce total job tardiness, e.g. in a situation where there is a late jobs waiting for a machine which is processing a non late job and the processing can be completed

within the overtime. In this case, processing the non late job in the overtime will allow the late job to be operated in the overtime for catching up.

- There is an overloaded workcentre, or type of labour, and there is a job which can be processed in the overtime by using the workcentre and/or labour.

Scheduling jobs for overtime is based on a given priority rule, or the timing plan in predictive schedule. Scheduling of overtime will be ended when:

- all the operationable late jobs can catch up, and all the operationable overloaded workcentre become non overloaded; or
- end of overtime.

5.4 Approaches for determination of job's ultimate priorities

As discussed in the above sections, both order and resource oriented factors are taken into account during job selection; in other words, a job's ultimate priority would be determined by coordination of these two types of factors.

By this control strategy, dispatching process has two steps, look-ahead and job selection. In the first step, all the candidate jobs will be in turn examined and information obtained about the consequences of its job selection. In the second step, decision will be made about which job should be assigned resources, i.e. decide ultimate priority of each candidate job.

Since it is impossible to formalise the job selection by an analytical optimal function, the key issue in this step is to use a heuristic approach to describe

different situations according to both order and capacity related information, and combination of the control policies suitable for the situation.

As discussed in section 5.2.5, four levels of shop floor situations can be identified according to due date related factors. The process of job selection can begin from classification of the shop status.

Table 5.1 summarises information and control policies used in different situations (X means that information is used).

The proposed control strategy is essentially a heuristic procedure to identify dominating constraints according to available information, and then choose appropriate methods for job selection. As described in chapter 2, these constraints include organisational goals, i.e. maintaining shop floor stability in this study and described by number of tardy jobs, total tardiness etc; resource availability such as machine availability, etc; preference constraints, and so on.

Goals are the dominating constraints in situations where due date objectives have been violated or are going to be violated; resource availability is the dominating constraints when a dispatching workcentre is overloaded or is a bottleneck; the timing plan in a predictive schedule is the dominant constraint when all the relevant jobs are early. The control policies described in the above sections, are the preferred constraints which determine control actions in different shop situations and conditions.

Figure 5.3, 5.4, 5.5, and 5.6 respectively outline the dispatching process in different situations. These flowcharts show major approaches in the coordination of order and capacity management related factors in the dispatching process, i.e. the ways by which dominating constraints (which can be order related or resource related) can be in turn identified and used for job selection.

Table 5.1: Information and control policies used in different situations

<i>Information</i>		<i>Situations</i>			
		<i>Late job</i>	<i>Critical job</i>	<i>Due start job</i>	<i>No adverse effect</i>
<i>Priority group</i>		X	X	X	X
<i>Consequences of a selection</i>	Added tardiness	X			
	Added tardy jobs	X			
	No. of affected critical jobs		X		
	No. of delayed due start jobs			X	
<i>Dispatching workcentres</i>	Overloaded	X	X	X	X
	Bottlenecks			X	X
	Downstream of breakdown machine				X
<i>Succeeding workcentres</i>	Waiting time	X		X	
	Status	X		X	X
	Bottlenecks				X
	Queue length			X	X
	Number of arriving jobs			X	X
	Remaining work			X	X
<i>Control policies</i>		TA	CJ	DSJ KBB LRA	KBB DM LRA
	<i>Priority rules</i>	ODD RW	ODD RW	ODD RW	RW

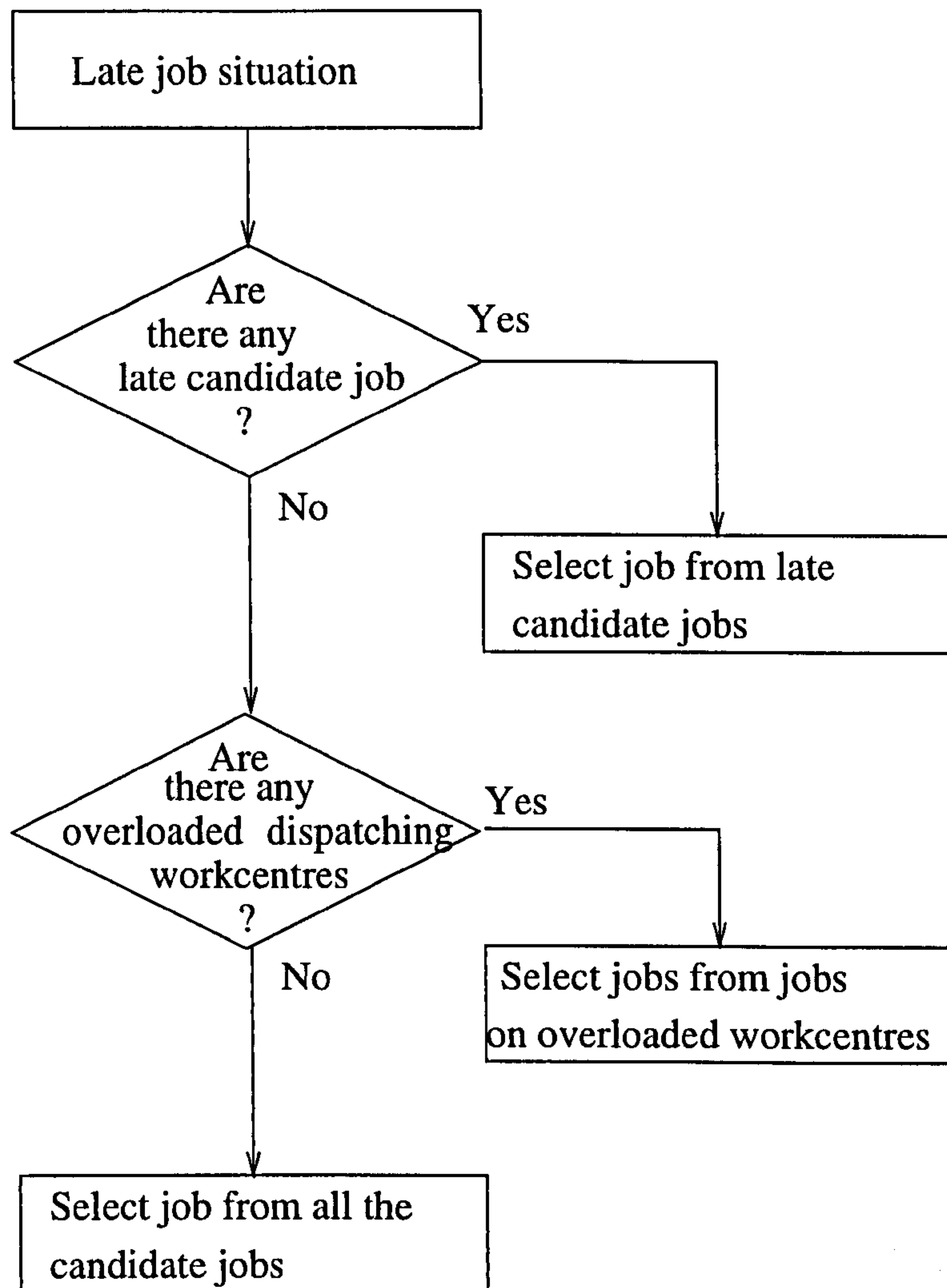


Figure 5.3: Process of job selection in late job situation

In a late job situation (i.e. when late jobs will inevitably result), as shown in **Figure 5.3**, any candidate job which is itself late, is given priority to be selected, and then overloaded dispatching workcentre(s), and so on. By these factors, a subset of candidated jobs (late jobs, jobs on overloaded dispatching workcentre(s), or all the candidated jobs) can be selected. The job selection will be carried out in these jobs by using a tardiness (TA) policy, as described in table 5.1. If there is more than one job in the group that causes equal minimum total tardiness, then the one with high priority ranking (by ODD) will be selected.

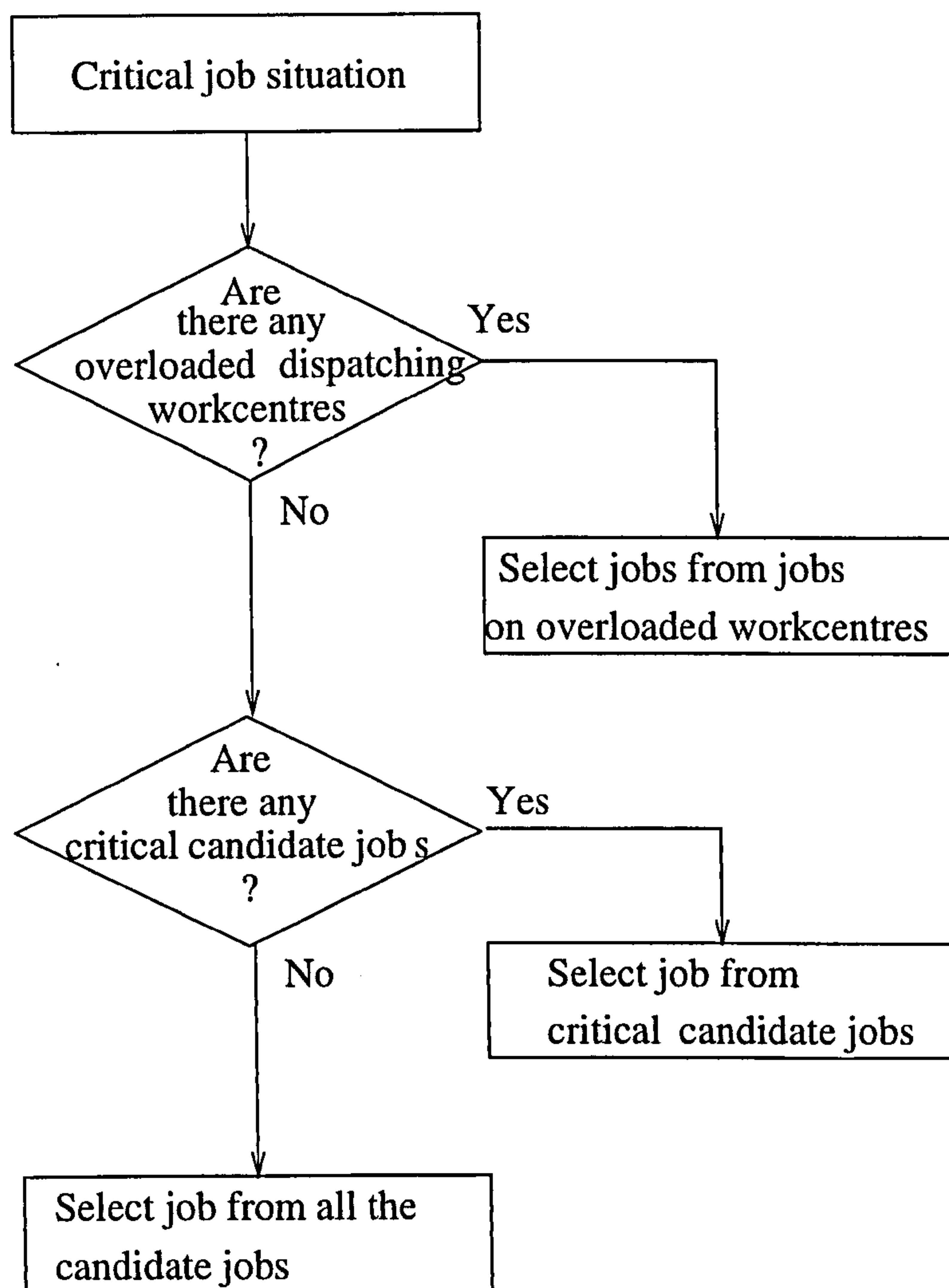


Figure 5.4: Process of job selection in critical job situation

In a critical job situation (i.e. when one or more critical jobs, candidate or non-

candidate, will be delayed), as shown in **Figure 5.4**, job selection would be based on the number of critical jobs affected. Jobs on overloaded workcentres will be given high priority since overloaded workcentres can lead to added total tardiness and/or increased overtime costs, and then any critical candidate job. If more than one job is chosen, then remaining work on overloaded workcentres (if there are any), and/or priority ranking (by ODD) of these jobs will be used as criteria for further decision making.

Remaining work on workcentres and the ODD can be seen as a means of a second level of decision making for ranking a set of jobs which are selected by some control policies but are in similar detected conditions, (e.g. same priority group and similar level of adverse effects). As shown in table 5.1, the ODD rule is used in late job, critical job or due start job situations; while remaining work on workcentres is used in all the situations.

In due start job situations, as shown in **Figure 5.5**, jobs on overloaded workcentres will be selected first, and then due start job will be selected if there is one. The other bottlenecks are taken into account later. Due start job policies are used for job selection in such a situation, i.e. if some due start jobs are selected by these factors, a look-road-ahead policy will be used, otherwise a non due start job will be selected by comparison with the number of due start jobs delayed.

In no adverse effect situations, as shown in **Figure 5.6**, jobs on overloaded workcentres will be selected first, the jobs on other bottlenecks second, and then jobs on workcentres which are at downstream of a broken down machine. When more than one job is selected by taking into account the above factors, information about their succeeding workcentres will be used for further decision making. As introduced in section 5.3.2, a machine idle rule will be used; a job for which the succeeding workcentre is a bottleneck, and is idle or going to be idle, will be given high priority. In this way, dispatching action tries to level work loading on both the dispatching workcentres and the direct downstream workcentres, i.e.

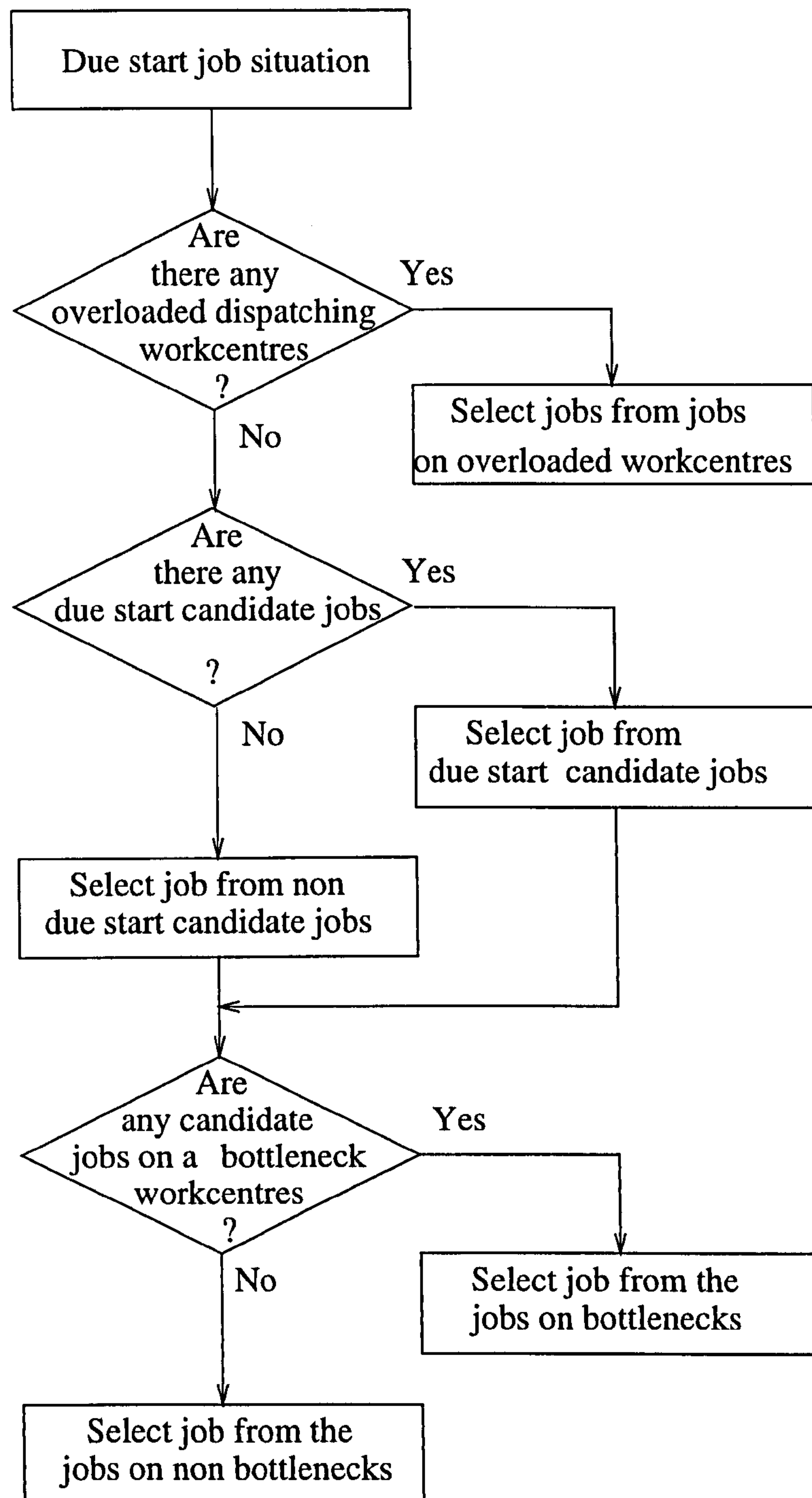


Figure 5.5: Process of job selection in due start job situation

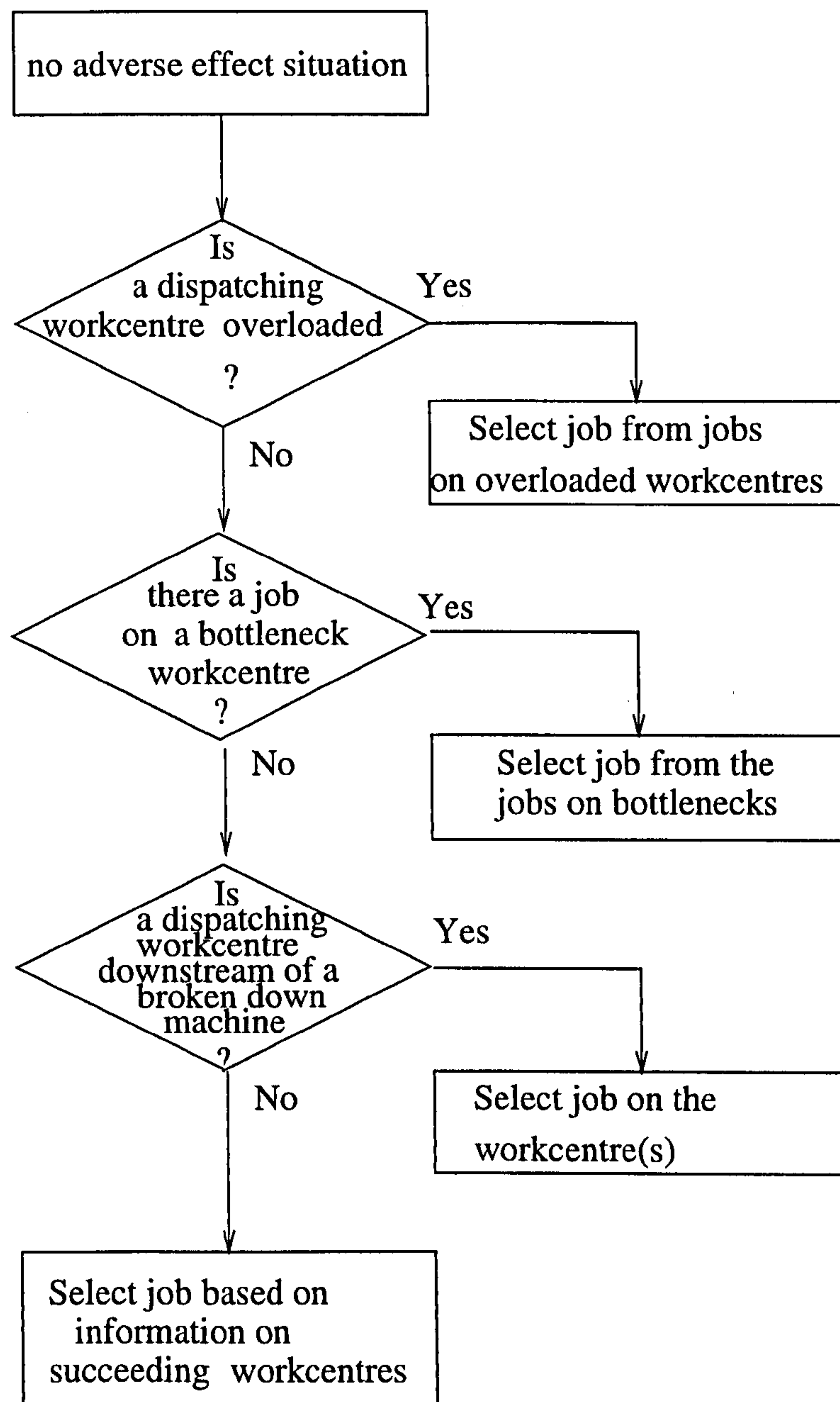


Figure 5.6: Process of job selection in no adverse effect situation

dispatching is capacity management oriented in these situations. Remaining work is used as the rule for secondary decision making.

In summary, by this control strategy, the criticality level of candidate jobs, described by priority group, is always taken into account first, unless some dispatching workcentres are overloaded. It can be seen that overloaded workcentre(s) is taken into account in all the situations except the situation of selecting a job from late jobs. But, bottlenecks (including long queue and potential long queue workcentres) are considered in the situations where there is no risk of violation of meeting due date or delayed critical job. This means that the imminent effect of job selection on the objective of control will be considered first. This is reflected in the way of job selection in the situations when added tardiness or delayed critical jobs are unavoidable.

For capacity management, load levelling among dispatching workcentre(s) is given higher priority than smoothing loading between the direct downstream workcentre(s) because the effect of load levelling on dispatching workcentre(s) is imminent, while that of on direct downstream workcentre(s) will have some time lag. Load levelling among the succeeding workcentres is only carried out in the situation of no adverse effects. The major approach for adjusting load level among the succeeding workcentres is 'Keep bottleneck busy, as introduced in section 5.3.2.

Chapter 6

Structure and implementation of a simulated manufacturing environment to test the proposed control strategy

6.1 Purpose of the simulated manufacturing environment

The objective of the simulated manufacturing environment(SME) is for analysing and understanding problems of maintaining shop floor stability, and for testing and comparing various shop floor control strategies, including priority rule based strategy and the proposed control strategy.

Due to the dynamic nature of the manufacturing environment, which is too complex to be effectively modelled by analytical approaches, discrete event simulation has become the most popular and appropriate approach for modelling and evaluating performance of manufacturing systems.

Simulation can be used in different phases of the analysis of a manufacturing process, ranging from system design to its operations. For real-time shop floor control, simulation is often used for designing and testing decision rules, (e.g. dispatching rules), or policy which can then be applied in real-time. [Waikar 1995] [Conway et al 1967]

Such a simulation-based, computerised mechanism needs the following components:

scheduler - for generating detailed, short term production plans, i.e. predictive schedules according to order due date requirements and available resource capacity. For maintaining shop floor stability, the generated schedules must be realistic in terms of no tardy jobs, and therefore, the scheduler should be a finite scheduling mechanism.

emulator - for the modelling physical structure of a manufacturing system and emulating material movement and operation of the equipments in the system. The emulator also needs to emulate random occurrence of disruption events.

control system for modelling the shop floor control strategy, which normally includes a resource allocation and a capacity control strategy at cell and/or system level, although it could include equipment control. Because of the complexity of decision problems on the shop floor (arising both from the combinatorial nature of the problem and the uncertainties arising from such sources as machine breakdown), shop floor control strategies often have to include unstructured, qualitative knowledge.

monitor/data collector - for monitoring the shop status, and translating and passing this information to the control system, and collecting data for statistics.

User interface - by which an user can establish, update system models which are going be tested, and conduct experiments and output results from experiments.

For testing and evaluating control strategies, the environment should be capable of representing heuristic knowledge in a way which is natural for shop floor personnel to express what they know, and allows easy modification of such knowledge (i.e. control strategies) without changing the overall structure of the program or the information about the program flow of control.

6.2 Review of conventional approaches to discrete event simulation

Research and development of simulation languages and related software has increased dramatically during recent years due to increasing demand, especially from manufacturing industries, and advancements in computing technologies.

Simulation languages and packages have been developed to help the users by simplifying the burden of handling repetitive tasks such as initialisation, time advancement, etc, and allowing them to focus their attention on using domain knowledge during the simulation process. For this, every simulation language or package must provide some commonly used functions, and define a way to express the logic of the model.

Two trends in the development of discrete event simulation for manufacturing systems can be identified: the software development trend, which aims to provide industrial users with more powerful, faster, more user-friendly tools; and the application development trend that brings about software being more 'system-friendly', i.e. integratable in a manufacturing environment, especially CIM.

6.2.1 Review of trends in development of conventional simulation approaches for manufacturing systems

From the 1980s, many simulation packages have provided user-friendly interfaces to make them easier to use and easier to understand. Currently, there are three types of commercial discrete event simulation packages available:

- 1) general-purpose simulation packages and languages, e.g. GASP IV [Pritsker 1974], SLAM [Pritsker et al 1979], SEEWHY [Istel 1987a] and SIMAN [Pegden 1985], were developed for general application. Therefore, the users are required to have the knowledge and skill to program the models by using general purpose computer language, e.g. Fortran, and/or the concepts defined in the simulation packages.
- 2) generic manufacturing simulators such as XCELL [John et al 1986], MAP/1 [Wortman & Miner 1986] and WITNESS [Istel 1987b]. In generic manufacturing simulators, the concepts and terminology used are oriented towards manufacturing, and these packages are intended to be suitable for a large range of discrete manufacturing applications.
- 3) specific types of manufacturing simulators, which have narrower terms of reference and model specific types of manufacturing systems, such as FMS, transport systems and so on , e.g. PROVISA [Marriott 1994] and Mast [Lenz 1985].

The last two types of simulation software, also called manufacturing simulators, are designed to be used by non-programmers and contain good user interfaces for users to think in manufacturing terms about the practical details of the system to be modelled. The users could be freed from any need to write a computer program, and have merely to learn the input specifications for the package, and any optional rules for the conduct of the model. In these simulation packages, interactive program generators may be used to generate the code of the simulation

program automatically, or a data-driven method is used to model system entries in data fields for specifying entities, activities, queues and so on.

However, manufacturing simulators, while very easy to use, lack flexibility and are also not fully transparent to the user in the way they actually work. As argued by [Adiga & Glassey 1991], manufacturing oriented simulators are not well-suited to research needs for shop floor control. As they pointed out these packages are either weak on representing shop floor control systems, e.g. MAP/1 and SIMFACTORY [Klein 1986]; or limited in the class of problems and decision rules they can handle. These packages are usually fixed in their view of the world. When complex shop floor control strategies need to be modelled, existing concepts and decision rules are insufficient, the user often must revert to lower level languages, e.g. Fortran, for representing the complex decision-making encountered in many real-world situations. When users have been forced down into the lower level language, the claimed advantage of the high-level simulation language is lost.

The use of graphics has also become an integral part of many simulation systems, and it can help to facilitate model definition and debugging as well as to display and help in the understanding of the simulation results.

There are three types of graphics applications in simulation [Carrie 1992]:

iconic use of 'iconic' elements to display the real system on the screen. This method is often used in manufacturing simulators, such as WITNESS.

logic- use of graphics in representation of the logical relationship between system components, e.g. in SLAM, SIMAN etc.

presentation - use of graphics for presenting and displaying output information and simulation results. The graphic display can be static, e.g. histograms, and dynamic such as animation.

In addition to the development of user-friendly simulation systems, there is also

an increasing effort to create open simulation software which can be functionally integrated and technically interfaced with other computer-based systems in a manufacturing environment. In such an application, the simulation is an integral part of the system, and can be used intensively in the operating phase of the manufacturing system.

A simulation system could be a part of the shop floor control system operating in an on-line on-going mode. With data collected from the physical production system and fed into a simulation model, the simulation could be used by the control system to evaluate decisions prior to their implementation. But, the relatively slow execution speed of simulation models and slow human decision making have inhibited developments of such real-time applications.

For efficient modelling and analysis of various shop floor control strategies, there is a need for a simulation tool which allows a detailed and easily adaptable approach for representation and changes in the control strategy, especially making use of the effective heuristic approaches used by shop floor personnel.

Because of the weakness of conventional simulation systems, [Shannon et al 1985] stated "What can be expected is the continuing enhancement and integration of existing fourth generation simulation languages and make them more powerful, and easier to use until they begin to bump into barriers imposed by underlying programming languages. At this point, either someone will devise a method of interfacing Lisp or Prolog-like languages to existing simulation languages or there will be the emergence of new, AI-based, fifth generation expert simulation systems based upon one of the recursive symbolic languages"

6.3 Knowledge based simulation system

Developments of knowledge based simulation systems arose from a desire to over-

come procedural weaknesses of conventional simulation systems and approaches. The primary one is the limitation of conventional language systems in the modelling and decision making process. Users have to be involved in many phases of the simulation process - designing the model, deciding upon a scenario (inputs), running the experiment, analysing the results, etc. Conventional simulation systems do not provide aid to the user in deciding upon an appropriate model or in how to exercise it to find answers to the problem being solved.

The objective of knowledge based simulation systems is to build into the modelling system most of the decisions that are now made by the simulation expert, and to make it possible for engineers, scientists, and managers to do simulation studies correctly and easily without elaborate training.

The power of knowledge based simulation partly comes from knowledge based technology, i.e. expert systems.

6.3.1 Review of important features of expert systems for shop floor control

“Expert systems are computerised problem-solving systems that can reach a level of performance comparable to that of human expert in some specialised problem domain.” [Shannon et al 1985]

One of the most important reasons for the presence of humans in shop floor control systems is their ability to adapt to abnormal system behaviour. It is often necessary for shop floor personnel to use past experience and knowledge to solve current problems. But, due to limitations associated with the processes of human decision-making and actions, errors can be made in the practice of shop floor control. The limitations can be scarce human expertise; limitation on human physical or mental ability, e.g. limited workload and working memory, unable to

quickly comprehend large amount of data, slow in recalling information stored in memory, etc; and even deliberate or inadvertent bias in their actions or avoidance of decision responsibilities.

Expert systems aim to offer an environment where the good capabilities of humans and the power of computers can be incorporated to overcome many of the limitations, and to fulfill the need for higher productivity and reliability of decisions.

The use of knowledge based approach for shop floor control, as reviewed in chapter 3, is an important area for the application of expert systems. The power of knowledge based systems is based on their architectures which allow computers to be able to retrieve and effectively use both factual and heuristic knowledge. A knowledge based system usually consists of:

Working memory as a global data base for storing the data for the problem under consideration and keeping track of the current solution status or situation,

Knowledge base consisting of problem solving knowledge associated with the problem domain. It describes the facts and heuristic knowledge in the form of rules, procedures and so on.

Inference engine a generic control mechanism which defines the problem solving approach or how the data and knowledge can be manipulated to solve the problem.

By separation of data, knowledge and control mechanisms, the knowledge based systems provide great flexibility in modelling, adding and updating of shop floor control strategies.

Expert systems are typically written in special programming languages, e.g. LISP,

PROLOG and OPS. Using these languages in the development of expert system ‘simplifies’ the coding process. The major advantage of these languages, as compared with conventional programming languages, is the simplicity of the addition, elimination, or substitution of new rules and memory management capabilities. Some of these languages have particular control algorithms built in, e.g. PROLOG has backward chaining, and OPS has forward chaining.

A rule based approach is often used for shop floor control.

6.3.2 Review of rule based approaches

In expert systems, rules, sometimes called production rules, are often used to encode empirical associations between patterns of information presented to the system and actions that the system should perform as a consequence.

Rules in a rule based system have the following general form:

$$P_1, \dots, P_m \rightarrow Q_1, \dots, Q_n$$

Where P_1 and ... and P_m are predicates which usually called *conditions* and/or *left-hand side* of a rule; Q_1 and ... and Q_n are *actions* which usually called *right-hand side* of a rule. If P_1 and ... and P_m are true then perform actions Q_1 and ... and Q_n .

The condition part of a rule is used to test the current state of the system described in the working memory, and the action then changes the current state. This could in turn give rise to new states that ‘produce’ more action (hence the name *production rule*), and so on until the system either reaches a solution or halts. In this way, rather than representing true statements about the problem or computing the values of functions defined over data, rules determine how

the symbol structures that represent the current state of the problem should be manipulated to bring the representation closer to a solution.

Rule-based systems have similar basic structure to other expert systems, including a rule base for storing the rules, a working memory for holding data, and an inference engine (usually called interpreter) for interpreting rules and selecting the next rule to apply.

Rule-based systems are specially suitable for real-time control problems because the problem of shop floor control is natural to be expressed in If-Then rules. It is more common and natural for shop floor personnel to express what they know as If-Then associations than as algorithms or all-inclusive theories.

The problem of shop floor control also requires a growing or changing knowledge base for testing and/or improving the effectiveness of control strategies. Rules are modular nuggets of information that are not explicitly directed by control statements in the program. So, it is possible to add or remove rules without changing the overall structure of the program or the information about the program flow of control.

The function of the interpreter (inference engine of a rule based system) can be described in terms of the recognise-act cycle, which consists of the following sequence of steps:

- 1) Match the condition (or left part) of rules against elements in working memory.
- 2) If there is more than one rule that could fire, choose one to apply; this step is called conflict resolution.
- 3) Apply the rule, perhaps adding a new item to working memory or deleting an old one, and then go to step 1.

There are two general approaches to conflict resolution: global control, which

tends to be domain-independent, and local control, which is usually domain-dependent. Global control strategies are usually “hard-coded” into the interpreter in commercial software packages, and therefore difficult for a programmer to change. Based on available global control strategies, users can use local control techniques, such as meta-rules to create particular effects for specific applications.

Since each production rule is self-contained, i.e. one rule never directly calls another, some rules called meta-rules can be used to direct the reasoning required to solve the problem rather than actually perform that reasoning. Meta-rules are usually domain-specific, though they could be domain-free.

Conflict resolution mechanisms vary from system to system, but three approaches refractoriness, recency, and specificity are very popular, and are often used in combination to form a global control regime.

Refractoriness means that a rule should not be allowed to fire more than once on the same data; recency refers to the strategy that rules that use more recent data are preferred to rules that match against data which has been loitering in working memory for some time; and specificity is a strategy of ranking rules by the number of conditions they have, i.e. rules that have a greater number of conditions and are therefore more difficult to satisfy, are preferred to more general rules with fewer conditions. The idea is that more specific rules are ‘better’ because they take more of the data into account.

The production rules paradigm can be encoded in non-rule based AI language, e.g. Lisp or Prolog, or embedded in languages such as OPS-5, but “the point is that that production rule languages, are specifically designed to do this, and as a result they do it rather well.” [Jackson 1992]

OPS-5 is a rule-based language, which allows researchers to capture the behaviour and expertise of shop floor personnel in software code and hence develop useful

scheduling and control tools. OPS-5 uses all three conflict resolution strategies to good effect.

Rules in OPS-5 have the form:

```
(p < rule - name >
< condition1 >
...
< conditionm >
- >
< action1 >
...
< actionn >)
```

The basic function of the working memory (WM) in OPS-5 is to hold data in the form of object-attribute-value vectors.

OPS-5 is essentially a forward-chaining language which uses the rules to move from a set of initial data to construct a solution. Scheduling and shop floor control problems can usually be solved by forward-chaining rules. The task of real-time control starts with the some jobs to be assigned resources and constraints on availability of resources, precedence of operations, etc, and reaches the decisions by heuristics (rules of thumb) that satisfy the constraints.

The most notable difference between OPS-5 and other programming languages is that the control of program flow in OPS-5 is not expressed in explicit control statements. The language is data-driven; the rule interpreter chooses the rule to execute depending on the data that match the rules.

In OPS-5, there are two alternative conflict resolution strategies, lexicographic-

sort (LEX) strategy, and the means-ends-analysis (MEA) strategy. Both LEX and MEA strategies have combined refraction, recency, and specificity criteria. But, LEX only uses the recency criteria on all condition elements of each rule, while MEA uses the recency criteria on first condition element, and then on all condition elements. Therefore, the MEA strategy can provide much tighter and global control during conflict resolution. The focus on the first condition element of each rule presents a natural opportunity to use that element for organising rules in different groups based on different subtasks.

OPS-5 can be a good tool for prototyping shop floor control strategies since users can concentrate less on control statements in the program and more on understanding and representing the knowledge of the shop floor control.

The power of knowledge based simulation systems come from the unification of expert systems and simulation technologies. Discrete event simulation is capable of modelling a manufacturing system in any required detail and generating a large number of scenarios (including random event, e.g. machine break down), which the shop floor control system may have to examine for making appropriate decisions, while the knowledge based systems provide the facilities of manipulating heuristic knowledge.

6.3.3 Approaches for implementation of knowledge based simulation systems in shop floor control

Expert systems and simulation may relate in different ways. Several different approaches have been described [O'Keefe 1986]. But, there are basically two different approaches:

hybrid systems - two separate pieces of softwares (simulation and expert system) are integrated and interfaced with each other in some ways,

new systems - changing the simulation modelling paradigm.

In a hybrid system, an expert system can be integrated and interfaced with a conventional simulation language or an existing manufacturing oriented simulator.

This approach is often suggested and used by researchers, e.g. [Ben-Arieh 1986] to problems of shop floor control, since they are relatively easy to develop, and the finished model executes at a fairly rapid speed. In such applications, an AI language, e.g. Lisp, Prolog, or OPS-5 is often used for modelling the control strategies.

But when an AI language is interfaced with a general purpose simulation system, users have to develop models by using a general purpose computer language, and/or the formats defined by the simulation system; when interfaced with a manufacturing simulator, there are some difficulties in communication between the two software packages.

Another approach is to create simulation systems in AI languages, like Lisp, Prolog, OPS-5, other than procedure oriented languages such as Fortran.

There are some researchers who have developed a simulation program for special purpose application in an AI language. For instance, [Shivnan & Browne 1986] used the language OPS-5 to create a knowledge-based simulator to use specifically for scheduling and real-time shop floor control. But, such systems are either developed for specific types of systems and not suitable for testing shop floor control strategies, or are not commercially available.

A major trend in this approach is to couple the AI and simulation methods within the same shell to provide an 'Integrated, Intelligent simulation environment'. This trend shows up in the availability of software products, e.g. Knowledge Craft [Carnegie 1988] and SimKit, [Intellicorp 1985] which include both expert system

and simulation programming tools. By using these tools for simulation, users are able to incorporate knowledge based techniques within simulation, giving users new possibilities in the process of modelling, simulation, and analysis. These systems often provide other facilities to aid in the development of applications, such as interactive user interface, graphic display, and so on.

The simulated environment developed for this research has been based on Knowledge Craft.

6.3.4 Overview of important features of Knowledge Craft for implementation of the environment

Knowledge Craft which is written in Common LISP is a tool kit for the development of expert systems. It allows the use of several knowledge base programming languages, CRL (Carnegie Representation Language), CRL-OPS, and CRL-PROLOG.

Knowledge Craft provides an integrated object oriented programming environment which allows natural modelling of the 'real-world' and helps to enforce information hiding and data abstraction, so that the user could readily model real systems in any level of detail. CRL provides the functions for manipulating (such as creating , deleting, and modifying schemata) objects, and their attributes and relational information.

CRL allows procedural functions to be associated with slots in the form of demons. Demons fire when an attempt is made to add, delete, or modify a value in a particular slot.

Context is a mechanism for the management of knowledge base version and alternate worlds reasoning. Contexts in CRL act as virtual copies of knowledge bases

in which schemata can be created, modified and destroyed without altering the original context.

CRL-OPS is based on OPS-5 with some additional functionality. The left-hand side of each CRL-OPS rule is described by schemata which are 'is-a' related to the class of the condition element. CRL-OPS rules can thus be written at the level of generality which is appropriate for the action of the rule. In this manner, rules may be written to apply either to a unique schema or a class of schemata. Rules will apply to schemata dynamically created at run time.

CRL-OPS is data-driven, which means that when there is a change in any schema referenced by a rule, the rule interpreter is automatically notified. In addition, CRL-OPS supports left-hand side functions, access to the CRL context mechanism, and integration with CRL-PROLOG. CRL-OPS rules can be compiled into an efficient run time form.

Knowledge Craft provides a tool-kit for discrete event simulation, SIMPAK, and allows the user to simulate the behaviour of a system defined with CRL. All SIMPAK modules have been designed using the object-oriented programming (OOP) paradigm. The components of each module are implemented as objects that contain methods which define their behaviour, e.g. events are defined as messages that are sent to objects at a specified time.

SIMPAK is composed of two modules, statistics and simulation. The statistics module provides random-number generators and probability distributions for representing statistical behaviour aspect of an application and generating output statistics. The Simulation module simulates the dynamic behaviour of an application during program execution. The Simulation module contains methods and functions for manipulating events on the calendar, and for creating, initialising, starting, and stopping a simulation.

Knowledge Craft also provides some tools to reduce users' programming efforts required to build application interfaces. The Window/graphics system provides the functionality to program graphic display for modelling and animation. By the Command system, menu interfaces are easy to build using a hierarchical command system that includes pop-up menus, mouse pointing, multi-word spelling completion, and help facilities.

On the whole, Knowledge Craft is an expert system development environment which provides an integrated set of tools for knowledge representation, reasoning and interfacing with end-users. Its integrated multiple programming styles can facilitate the use of heuristic or non-procedural knowledge and simulation techniques in shop floor control.

6.4 Structure and implementation of the Simulated Environment

As shown in **Figure 6.1**, the Simulated Manufacturing Environment consists of a schedule generator, control module, monitor, a shop floor emulator and an user interface.

The schedule generator is a finite scheduling mechanism, and is used for generating a detailed production plan. It takes as its input the planned orders for a short period of time, such as two weeks, and proceeds to develop a plan for all of the operations through their respective workstations, i.e. specifying the timing of operations in order to comply with due dates, availability of machine and labours, etc. The schedule generator then provides the time sequenced operation plan to the shop floor emulator for execution. A set of priority rules, such as EDD, ODD, CRR, are provided by the system for the schedule generator to schedule the flow of work through the shop floor.

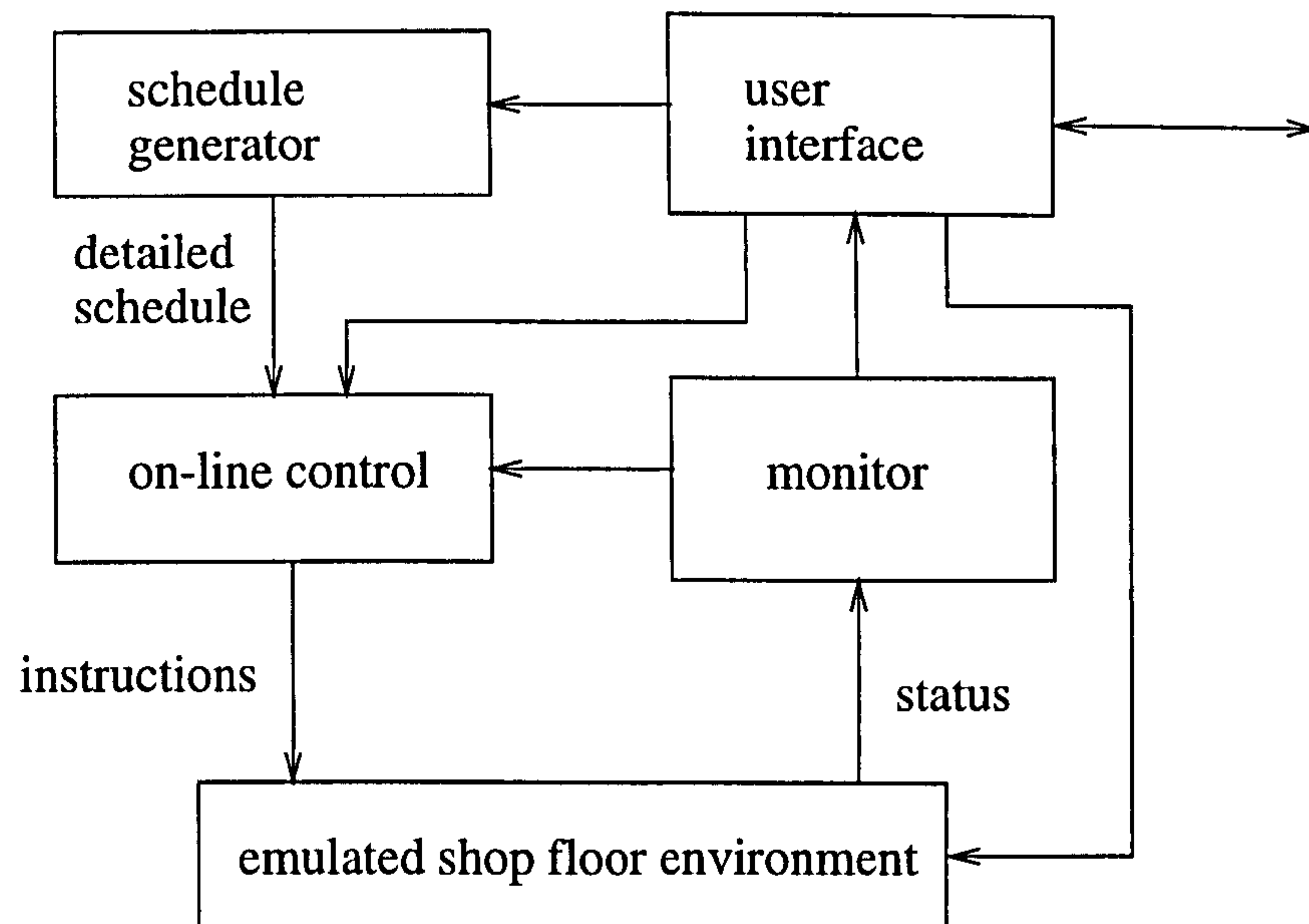


Figure 6.1: The structure of the Simulated Manufacturing Environment

The shop floor emulator simulates all of the events that occur on the shop floor, parts movement, operations, and unexpected events, e.g. machine breakdown.

The monitor performs the real time data collection and feedback functions. It collects data on current shop status, machine status and labour availability, progress of jobs, current loading and remaining work on each workstation, etc and reports them back to the appropriate module within the system. This information is passed to on-line control for dispatching and overtime decisions, as well as to the user interface for showing current shop status in the form of text and graphic display, and output statistics.

The control module includes a dispatcher for dispatching jobs, and a capacity control facility for decision-making about using overtime. Decisions made in this mechanism will be translated into instructions to activate corresponding actions in the shop floor emulator.

The user interface is for modelling the manufacturing systems studied and output

the results. By means of the interface, the manufacturing system model can be established, the scheduling rules can be selected for generating a predictive schedule by schedule generator, and control strategies can be changed or selected for on-line control, and shop status and statistics can be outputted.

The Simulated Manufacturing Environment (SME) is developed by using the expert system shell, Knowledge Craft.

6.4.1 Structure and implementation of the simulation system in SME

An objected-oriented approach is used for the development of the simulation system. Objected-oriented programming treats a program as a collection of objects. An object-oriented simulation system would contain three types of objects: domain independent, domain dependent, and application specific.

Domain independent objects are the objects which are common to and needed by all simulation models. They provide behavioural definitions for a generic set of model components such as random number generators, statistical analysis modules, etc. Domain dependent objects describe the model components that correlate to real components of the system which would be used in a particular application, but are general to the domain of interest. For example, a manufacturing simulation system would have pre-defined objects for workers, machines of different types, material handling systems, etc. These domain dependent objects provide the templates for the creation of specific instances of the object described. Application specific objects provide information on the specific combinations and numbers of components needed for the specific study that is being undertaken, as well as the sequence of model components that are activated during the execution of the model.

By this way of modelling, the object-oriented approach provides a flexible, extendible system and permits objects to be dynamically re-configured.

The simulation model in the environment is a collection of manufacturing oriented objects (schemata). As shown in **Figure 6.2**, the environment consists of several types of objects: physical objects for modelling its physical structure, materials, human operators, etc; **shift** for describing shift system used in the shop, schedule related objects for modelling a schedule, and a shop floor control system for modelling control activities.

Machine, **buffer**, **part** and **labour** have been defined for modelling physical elements and structure of a manufacturing system. As shown in **Figure 6.3**, slots of these objects are used to represent facts and behaviour about the corresponding real world entity of the objects. The facts include static as well as dynamic data. Static data is used for describing inherent features of the object, (e.g. the slots on the object **machine** concerned with breakdown, set up and its other features), while dynamic data represents current information about the object, such as the status of a machine.

Jobs (schema name is **sme-job**) and operations are the objects for modelling the production plan. It can be seen from **Figure 6.4**, some of the slots in these objects are for storing manufacturing information, e.g. batch-size and part-number in job objects, input-parts and output-parts slots on operation objects; some for representation of a predictive schedule, e.g. planned-start-time; and some for describing current status of the schedule, i.e. progress of each job, such as next-operation and lateness.

The shift system can also be modelled by using object oriented approaches. There are two types of shift objects in the system, normal shift and overtime shift. The

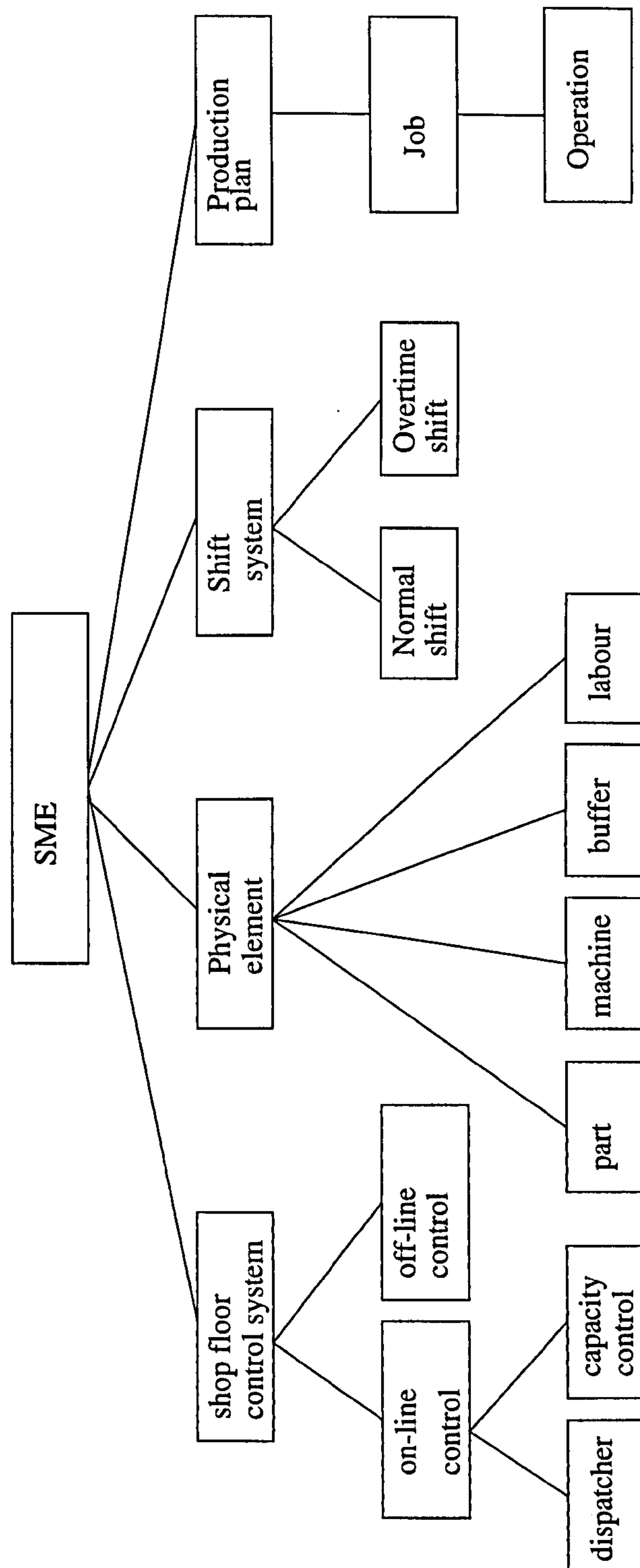


Figure 6.2: Components of the Simulated Environment

```

{{ MACHINE
    IS-A PHYSICAL-ELEMENT
    NAME
    TYPE
    QUANTITY
    PRIORITY
    BREAKDOWN
    BREAKDOWN-TIME-BASE
    TIME-BETWEEN-FAIL
    REPAIR-TIME
    INPUT-BUFFER
    OUTPUT-BUFFER
    OPERATIONS
    CURRENT-OPERATION
    CURRENT-LABORS
    STATUS
    CURRENT-REMAINING-WORK
    MAXIMUM-REMAINING-CAPACITY
    MINIMUM-REMAINING-CAPACITY
    ARRIVING-JOBS
    PARTS
    .....
    INPUT
    OUTPUT
    ELEMENT-ARRIVE
    FINISH-MACHINING
    BREAKDOWN
    FINISH-REPAIRING
    .....
}}

{{ BUFFER
    IS-A PHYSICAL-ELEMENT
    NAME
    TYPE
    QUANTITY
    OUTPUT-TO
    JOBS
    PARTS
    .....
    INPUT
    OUTPUT
    ELEMENT-ARRIVE
    .....
}}

{{ LABOR
    IS-A PHYSICAL-ELEMENT
    NAME
    TYPE
    QUANTITY
    STATUS
    CURRENT-POSITION
    ON-SHIFT
    .....
}}

{{ PART
    IS-A PHYSICAL-ELEMENT
    NAME
    TYPE
    QUANTITY
    JOBS
    .....
    OUTPUT
    }}

```

Figure 6.3: Structure of physical element schemata

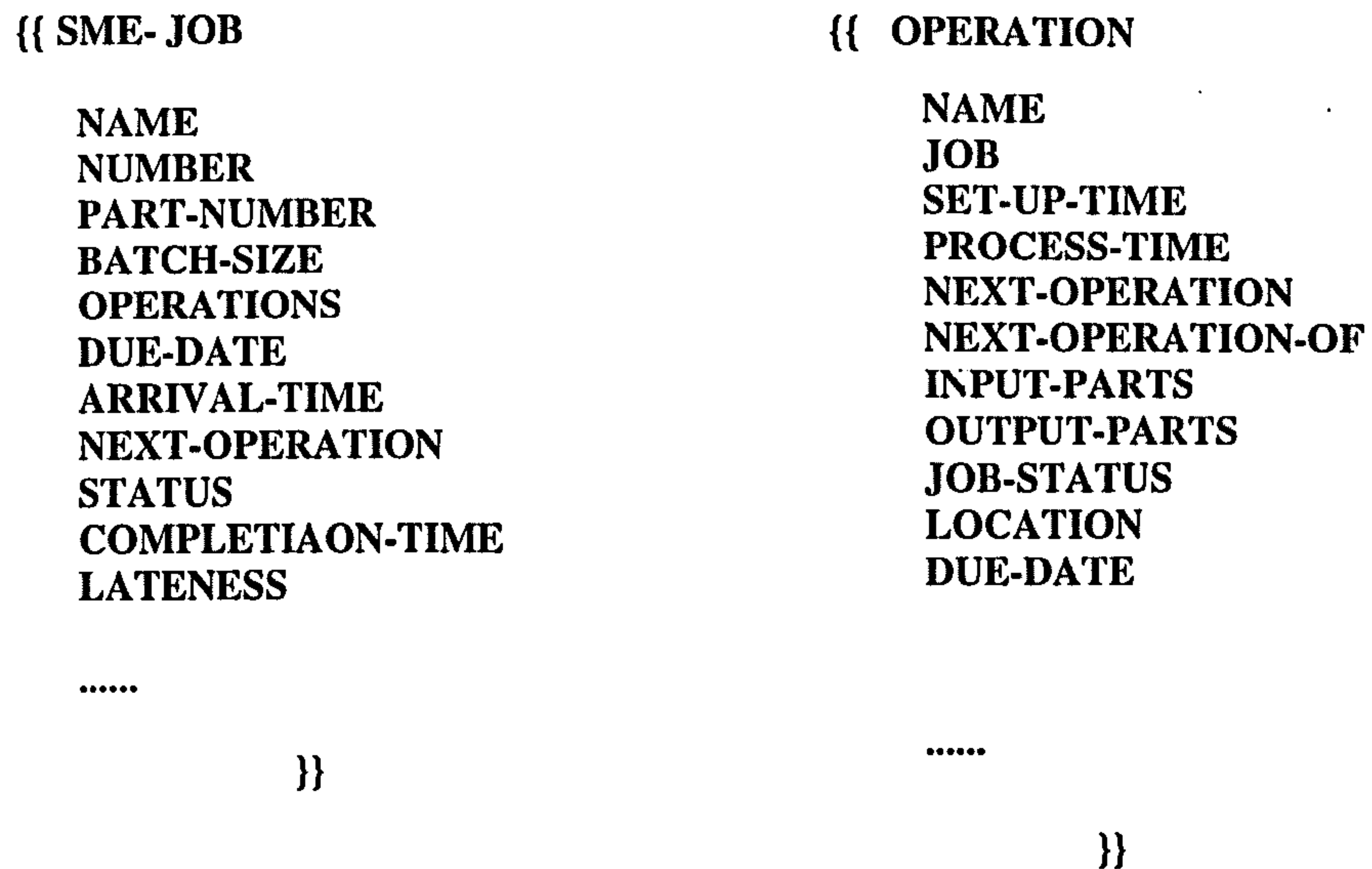


Figure 6.4: Structure of job and operation schemata

normal shift is the shift for which resource capacity (machine and labour) has been taken into account when generating the predictive schedule. Overtime shift describes information about the time gaps which are between normal shifts and within which overtime production is allowed, i.e. in which period, resources are available for operating jobs. Shift schema is shown in Figure 6.5. With a defined shift mode, the system can maintain a real time clock, and many activities in the system can be based on this real time clock, such as checking job due dates, overtime start time and end time, etc.

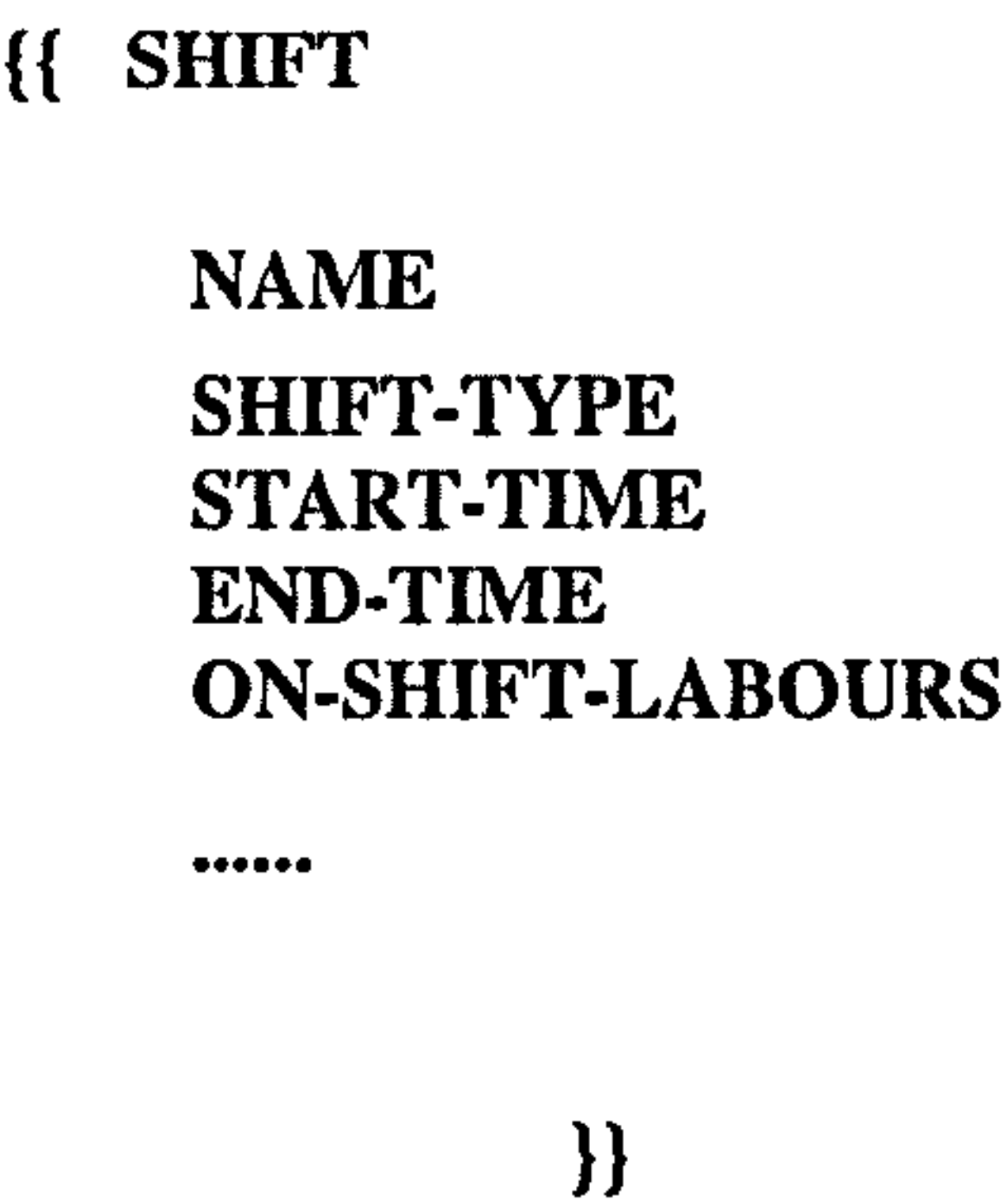


Figure 6.5: Structure of shift schema

The relationship between those objects can also be described through slots.

Slots can be used to describe the physical structure of manufacturing systems, e.g. input-buffer slot on a machine and input-buffer-of slot on a buffer are for describing a permanent physical link between a machine and a buffer; the input-buffer slot of a machine is for recording the name of its entry buffer. Some slots are for describing a temporary relation of objects, e.g. current-operation on a machine object, current-position of a worker, etc.

Some slots are for storing the methods which describe actions the object is to execute when sent the appropriate messages, i.e. describe behaviours of those objects. For example, output slot on a machine stores the method which unload finished parts from the machine.

A simulation model can be established by creating required, application specific objects from these physical modelling objects. The created objects can inherit information (both facts and methods) from their ancestors, and the model is configured from information on the specific combinations and numbers of components needed for the study as well as the specific instances of methods that are to be activated during the execution of the model.

Jobs are released to the system by given arrival times in a plan and would pass through each step of the operation by planned routings. SIMPAK provides the functions to manage the simulation calendar and clock. A simulation event is associated with a method which is referenced by objects and messages, instead of functions. At each event time, SIMPAK would activate the methods which relate to current simulation events. The control module is activated at each event time after all the current simulation events have occurred (three phase simulation).

The control module is represented by CRL-OPS rules and some procedural program. Dispatching policies can be described by either a priority rule or CRL-OPS based control strategies, and overtime is described by a procedural program.

A menu-driven approach is used in the user interface for the creation of models,

and there are three stages for developing a model: define, display and detail. Like WITNESS, all the elements in the system to be modelled are defined in the define stage; display are set up in the display stage; and more detailed information are inputted at the detail stage. This environment also allows all the modelling information be inputted through a file in a given format.

The simulation model can be used as a schedule generator for generating a detailed production plan by using a due-date based priority rule chosen from a priority rule library in the system; or a shop floor emulator for testing control policies.

A detailed production plan can be generated according to due date requirements of the jobs and available resources in the scheduling horizon. After sequences of operations on each workcentre, and a timing plan of resource assignment to each operation are determined, the plan is recorded.

When a realistic schedule has been generated, it can be used for testing control strategies represented by a priority rule or CRL-OPS rule based system. Parameters about machine breakdown, e.g. time of first breakdown, time interval between breakdown, repair time, etc. can be specified through the interactive user interface. Simulation can be run in two modes, *execute* which means the generated plan will be executed until all the jobs have been completed, and *advance* which allows the simulation to stop at any time point during the execution. The former mode is for statistical analysis of simulation results, while the latter for generation of various shop floor scenarios for demonstration and comparison of shop floor control strategies.

The output of the simulation result has two forms, dynamic and static. Dynamic is for output of current model status by means of text and graphic display, and static is for print out of statistics.

The graphical display and animation are implemented by means of a Graphics

system in Knowledge Craft. A set of icons are provided for showing physical (spatial) relationships between system components and animating the flow of parts (jobs) through the system. The components of the system modelled can be displayed by these icons with user chosen size at user specified positions on the screen. Each physical element can be displayed in different colours. The colour of a machine indicates its status, i.e. running, waiting, blocked, broken down, or being set up. This can be very useful in debugging (verification) and validation of the simulation by showing whether the results are logical and the model is behaving like the real system. Four windows are available for displaying model elements, and the user can use the menu and mouse to decide which window, and where and how a specific object should be positioned. When display of a model element is described through the user interface (either interactive or inputted through file), a corresponding display item will be created and linked to the element, i.e. name of the item is stored in a particular slot of the model element. When the display of the model element needs to be changed, such as colour, the system will find the display item and make the changes.

Demons have been used in the system to activate the procedures which are responsible for maintaining and manipulating the model display. For example, a demon is used to monitor access of slot 'current-operation' on each machine. When a job is loaded onto or unloaded from a machine, i.e. its next operation is being put into or deleted from 'current-operation' slot of the machine, the operation name can be displayed or erased from the screen.

Animation and graphic display can be switched off when testing control strategies for speeding up the simulation execution.

Statistics of the simulation results include total tardiness, number of late jobs, and overtime manhours. Total tardiness and number of late jobs are the most commonly used measurement for due date performance. Overtime manhours is used for measurement of extra production cost. For maintaining shop floor

stability, both performance and cost measurement needs to be taken into account. The system would generate the objects which are subclasses of the relevant objects in SIMPAK for data collection and statistics, and link these objects to specific objects in the physical model.

The simulation model can not only simulate normal shift production process but also overtime. The user defined shift model, described by shift schemata, provides information about available capacity of machines and labours in the schedule time span (in normal shifts), and potential capacity of machines and labours in the time gaps between normal shifts. At the end of a shift, the control module will check if overtime is needed and if overtime is available. If so, the control module will decide which operations should be operated in the overtime and which machines and labour are required, and then this plan will be implemented during the overtime. If not, the simulation clock simply advances to next normal shift's start time.

The major difference between simulation of normal shift production and overtime production is that in a normal shift all the resources, machines and labours, are available, and operation on jobs is continuously carried out; in an overtime (shift), however, some of the resources may not be available, and some operations will stop due to lack of resources (in most situation, due to lack of labour) and restart at the start time of the next normal shift.

Implementation of simulation of overtime is illustrated as follows:

- 1) At the stage of model development, overtime shifts can be defined and detailed, which store information concerning start time and end time of each overtime, available labour, etc.
- 2) When the simulation clock advances to a time point (normally end of shift and/or end of day) at which an overtime is available, check if conditions of overtime are met. If conditions of overtime are met, i.e. overtime is

required and some jobs can be processed in the overtime, go to 3) otherwise go to 5)

3 The control module decides which operations will be operated in the overtime, (called overtime operations), and then advances simulation clock to next event time.

4) Implementation of overtime plan until end of overtime.

5) Simulation clock advances to beginning of next normal shift.

The condition for the use of overtime and decisions about overtime operations represent the capacity control policy in the system, and is included in the control module as part of its control policy.

6.4.2 Control module

The control module includes two submodules, dispatching and overtime.

Dispatching submodule is used for execution of dispatching by a priority rule or the proposed control strategy, and activated at a simulation event time and after all the current simulation events have occurred.

The dispatching submodule first checks if and how many jobs can be selected according to the current shop status. Three situations and related control activities can be described as follows:

- 1.** when there is no candidate job, no control action,
- 2.** when there is one candidate job, or the candidate jobs do not compete for the same types of machine and labour, assign the machine and labour to all the candidate jobs,

3. when there are more than one candidate jobs which demand the same types of machine and/or labour, select the job according to the control strategy set in the dispatching submodule, which is either the proposed control strategy, or a priority rule.

The priority rule based control is implemented by procedural program while the proposed control strategy by the rule-based language CRL-OPS.

In the CRL-OPS rule based dispatching system, MEA conflict resolution strategy is used for deciding the next rule to apply. The CRL-OPS rules which describe the proposed control strategy, are organised in several groups according to the first element of each rule.

There are two schemata, control-context and control-action, defined in the system for conducting of the dispatching process **Figure 6.6**.

{{ CONTROL-CONTEXT	{{ CONTROL-ACTION
IS-A OBJECT	IS-A OBJECT
GROUP	OPERATION
DISPATCHING-WORKCENTRES	JOB
BOTTLENECKS	JOB-GROUP
CURRENT-LATE-JOBS	LATE-JOBS-CAUSED
SCHEDULABLE-OPERATIONS	INCREASED-TOTAL-TARDINESS
AVERAGE-QUEUING-TIME	CRITICAL-JOBS-AFFECTED
.....	DUE-START-JOBS-AFFECTED
	PRIORITY
	LOCATION
	NEXT-SUCCESSOR-WORKCENTRE
}}	JOBS-AT-NEXT-QUEUE
	JOBS-ARRIVING-NEXT-QUEUE
	OPERATING-STATUS-OF-NEXT-WORKCENTRE
	WAITING-TIME-AT-NEXT-WORKCENTRE

	}}
(A)	(B)

Figure 6.6: Schemata used in control module

The control-context is for partitioning of production rules by functions and storing some information for dispatching. There are two types of rules in the rule base, meta-rules and normal rules. The meta-rules are for identifying the situations and deciding which group of rules are to be needed. The other rules are organised in rule groups for job selection in different situations.

The procedure of identifying the situations is showed in Figure 5.2, and four groups of rules can be used for dispatching jobs in different situations. Within each group, the rule selection retains purely data-driven behaviour, i.e. choosing next firing rule from the same group is based on criteria similar to the LEX strategy.

The schema **control-action** is defined for creating specific (instance) actions at a time point when dispatching is needed. These schemata would be used for recording information required for job selections. Each schema will record information concerned with selection of a particular job (operation), and the production rules would derive a solution based on this information.

For the sake of efficiency, all the information needed for dispatching will be recorded onto these instance schemata, so that the working memory does not need to store a large amount of data which will not be used in dispatching, such as operation related information.

Figure 6.7, Figure 6.8, Figure 6.9 and Figure 6.10 respectively show examples of CRL-OPS rules in different rule groups.

There is a capacity control strategy in the control module for using overtime to compensate capacity losses and to meet due date requirements.

There are a variety of constraints associated with the application of overtime. It includes constraints on the length of the overtime period and the time at which


```

(P SELECE-JOB-BY-TOTAL-TARDINESS

  (CONTROL-CONTEXT ^SCHEMA-NAME <CONTEXT>
    ^GROUP LATE-JOB
    ^CURRENT-LATE-JOBS { <> () <LATE-JOBS> } )

  (CONTROL-ACTION ^SCHEMA-CONTEXT <CONTEXT>
    ^INSTANCE CONTROL-ACTION
    ^OPERATION { <> () <SELECTED-OPERATION> }
    ^JOB (MEMBER <> <LATE-JOBS> )
    ^INCREASED-TOTAL-TARDINESS <TARDINESS>
    ^OPERATING-STATUS-OF-NEXT-WORKCENTRE <> UNDER-REPAIRING
    ^ESTIMATED-WAITING-TIME-ON-NEXT-QUEUE < WAITING-TIME > )

  - (CONTROL-ACTION ^SCHEMA-CONTEXT <CONTEXT>
    ^INSTANCE CONTROL-ACTION
    ^OPERATION { <> () <> <SELECTED-OPERATION> }
    ^JOB (MEMBER <> <LATE-JOBS> )
    ^OPERATING-STATUS-OF-NEXT-WORKCENTRE <> UNDER-REPAIRING
    ^ESTIMATED-WAITING-TIME-ON-NEXT-QUEUE < WAITING-TIME-1 >
    ^INCREASED-TOTAL-TARDINESS
    (LIST-SUM< 2 <WAITING-TIME-1> < WAITING-TIME > < TARDINESS > ))

  ->

  (NEW-VALUE <CONTEXT> 'SCHEDULING-OPERATION <SELECTED-OPERATION> :CONTEXT '$ROOT-CONTEXT ))

```

Figure 6.7: A rule for selecting late job by total tardiness

```

(P SELECT-CRITICAL-JOB
  (CONTROL-CONTEXT ^SCHEMA-NAME <CONTEXT>
    ^GROUP CRITICAL-JOB
    ^SCHEDULABLE-OPERATIONS <CANDIDATE-JOBS> )
  (CONTROL-ACTION ^SCHEMA-CONTEXT <CONTEXT>
    ^INSTANCE CONTROL-ACTION
    ^OPERATION { <> 0 <SELECTED-OPERATION> }
    ^JOB-GROUP CRITICAL-JOB
    ^INCREASED-TOTAL-TARDINESS { <= 0 }
    ^CRITICAL-JOBS-AFFECTED <CRITICAL-JOBS>
    ^PRIORITY <PRIORITY> )
  - (CONTROL-ACTION ^SCHEMA-CONTEXT <CONTEXT>
    ^INSTANCE CONTROL-ACTION
    ^OPERATION { <> 0 <SELECTED-OPERATION> }
    ^JOB-GROUP CRITICAL-JOB
    ^INCREASED-TOTAL-TARDINESS { <= 0 }
    ^CRITICAL-JOBS-AFFECTED (COMPARE-QUEUE-LENGTH< <> <CRITICAL-JOBS> ))
  - (CONTROL-ACTION ^SCHEMA-CONTEXT <CONTEXT>
    ^INSTANCE CONTROL-ACTION
    ^OPERATION { <> 0 <SELECTED-OPERATION> }
    ^JOB-GROUP CRITICAL-JOB
    ^INCREASED-TOTAL-TARDINESS { <= 0 }
    ^CRITICAL-JOBS-AFFECTED (COMPARE-QUEUE-LENGTH= <> <CRITICAL-JOBS> )
    ^PRIORITY {< <PRIORITY> } )

-->
(NEW-VALUE <CONTEXT> 'SCHEDULING-OPERATION <SELECTED-OPERATION> :CONTEXT '$ROOT-CONTEXT) )

```

Figure 6.8: A rule for selecting critical job

```

(P SELECE-DUE-START-JOB-WITH-SHORT-WAITING-TIME-AT-NEXT-QUEUE
  (CONTROL-CONTEXT ^SCHEMA-NAME <CONTEXT>
    ^GROUP DUE-START
    ^SCHEDULABLE-OPERATIONS <CANDIDATE-JOBS>
    ^AVERAGE-QUEUEING-TIME <AVERAGE-QUEUEING-TIME> )
  (CONTROL-ACTION ^SCHEMA-CONTEXT <CONTEXT>
    ^INSTANCE CONTROL-ACTION
    ^OPERATION { <> () <SELECTED-OPERATION> }
    ^JOB-GROUP DUE-START
    ^PRIORITY <PRIORITY>
    ^OPERATING-STATUS-OF-NEXT-WORKCENTRE <> UNDER-REPAIRING
    ^ESTIMATED-WAITING-TIME-ON-NEXT-QUEUE { <= <AVERAGE-QUEUEING-TIME> } )
  - (CONTROL-ACTION ^SCHEMA-CONTEXT <CONTEXT>
    ^INSTANCE CONTROL-ACTION
    ^OPERATION { <> () <> <SELECTED-OPERATION> }
    ^JOB-GROUP DUE-START
    ^OPERATING-STATUS-OF-NEXT-WORKCENTRE <> UNDER-REPAIRING
    ^ESTIMATED-WAITING-TIME-ON-NEXT-QUEUE { <= <AVERAGE-QUEUEING-TIME> }
    ^PRIORITY { < <PRIORITY> } )

-->

(NEW-VALUE <CONTEXT> 'SCHEDULING-OPERATION <SELECTED-OPERATION> :CONTEXT '$ROOT-CONTEXT))

```

Figure 6.9: A rule for selecting due start job by LRA control policy

```

(P SELECE-NON-DUE-START-JOB
  (CONTROL-CONTEXT ^SCHEMA-NAME <CONTEXT>
    ^GROUP NON-DUE-START
    ^DISPATCHING-WORKCENTRES (QUEUE-LENGTH=< 1)
    ^BOTTLENECKS <BOTTLENECKS> )
    (CONTROL-ACTION ^SCHEMA-CONTEXT <CONTEXT>
      ^INSTANCE CONTROL-ACTION
      ^OPERATION { < 0 <SELECTED-OPERATION> }
      ^JOB-GROUP NON-DUE-START
      ^INCREASED-TOTAL-TARDINESS { <= 0 }
      ^CRITICAL-JOBS-AFFECTED < 0
      ^DUE-START-JOBS-AFFECTED < 0
      ^OPERATING-STATUS-OF-NEXT-WORKCENTRE <  UNDER-REPAIRING
      ^JOBS-AT-NEXT-QUEUE 0
      ^JOBS-ARRIVING-NEXT-QUEUE 0
      ^NEXT-SUCCESSOR-WORKCENTRE (MEMBER < <BOTTLENECKS> ))
      - (CONTROL-ACTION ^SCHEMA-CONTEXT <CONTEXT>
        ^INSTANCE CONTROL-ACTION
        ^OPERATION { < 0 <SELECTED-OPERATION> }
        ^JOB-GROUP NON-DUE-START
        ^INCREASED-TOTAL-TARDINESS { <= 0 }
        ^CRITICAL-JOBS-AFFECTED < 0
        ^DUE-START-JOBS-AFFECTED < 0
        ^OPERATING-STATUS-OF-NEXT-WORKCENTRE <  UNDER-REPAIRING
        ^JOBS-AT-NEXT-QUEUE 0
        ^JOBS-ARRIVING-NEXT-QUEUE 0
        ^NEXT-SUCCESSOR-WORKCENTRE (MEMBER < <BOTTLENECKS> ))
      -->
      (NEW-VALUE <CONTEXT> 'SCHEDULING-OPERATION <SELECTED-OPERATION> :CONTEXT '$ROOT-CONTEXT'))

```

Figure 6.10: A rule for selecting non due start job

overtime can be applied, availability of resources, routing of jobs, etc. Whenever the simulation clock advances to a point of time where overtime is available, the overtime module is activated after all the current simulation events have occurred if there are some.

Conditions to use the overtime and its constraints will be checked in the following steps:

- 1) check if there is a late operation, or an overloaded workcentre (on which remaining work is over available capacity in the plan time span),
- 2) check if there is a late operation that can be operated during the overtime, or there is a job which can be operated on a overloaded workcentre.

If both of the conditions can be met, the overtime based capacity control action will be activated. All the executable late operations and the operations on overloaded workcentres will be scheduled for processing in the overtime. The scheduling is based on a predetermined overtime strategy that is either the operation sequence defined in the original schedule, or a priority rule, e.g. ODD. As described in chapter 5, scheduling will be ended if either of the above conditions can not be met, or at the end of the overtime.

The overtime scheduler is essentially a forward scheduling mechanism which can generate a detailed overtime operation plan. **Figure 6.11** outlines the scheduling procedure.

As shown in the figure, following jobs will be chosen as candidate overtime jobs for scheduling at the beginning of an overtime period:

- Late jobs which are currently loaded on machines, and late jobs which are currently waiting for machine and/or labour but can be operated in the overtime, (i.e. the required machine and/or labour will be available within the overtime period).

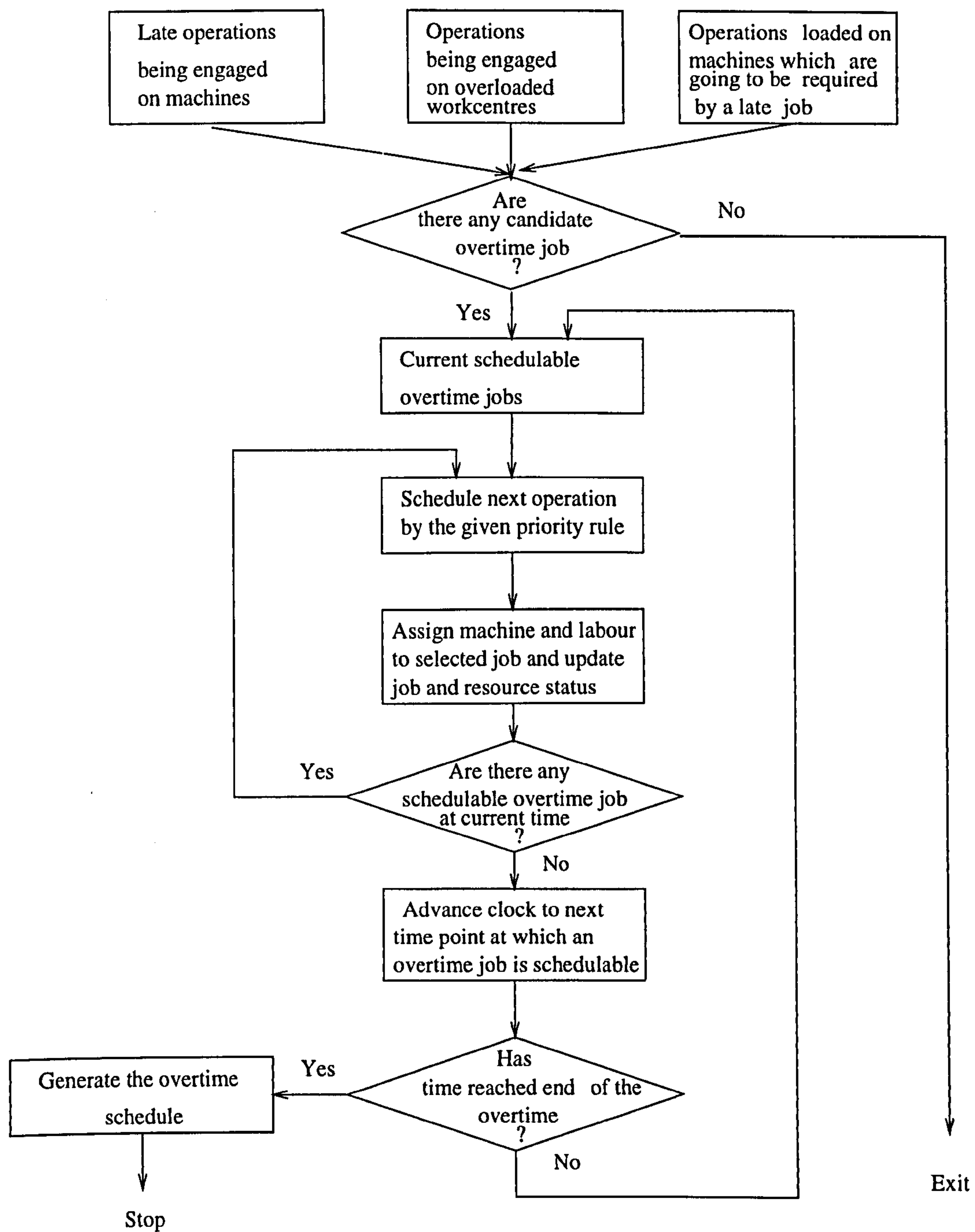


Figure 6.11: Process of overtime scheduling

- Jobs which are currently engaged on and could be operated on overloaded workcentres,
- Jobs which are currently on a machine which is currently or going to be required by late jobs within the overtime period.

Overtime scheduling will start if there is a candidate overtime job. The operations of the overtime jobs will be scheduled at the time point at which the required machine and labour are available. At each time point, job selection will be based on ODD priority rule. After assigning machine and labour to a job, some candidate overtime jobs may be removed from the overtime candidate job list due to changed shop conditions, e.g. a job has caught up with its schedule, a workcentre is no longer overloaded, etc. The scheduling will stop if there is no overtime job schedulable within the overtime period. A machine breakdown within the period may also lead to some jobs having to be taken off the list. Overtime production will follow this procedure until end of the overtime period.

An interactive, menu driven user interface is set up for management of the environment, including modelling and execution. **Figure 6.12** shows main structure of the menu hierarchy.

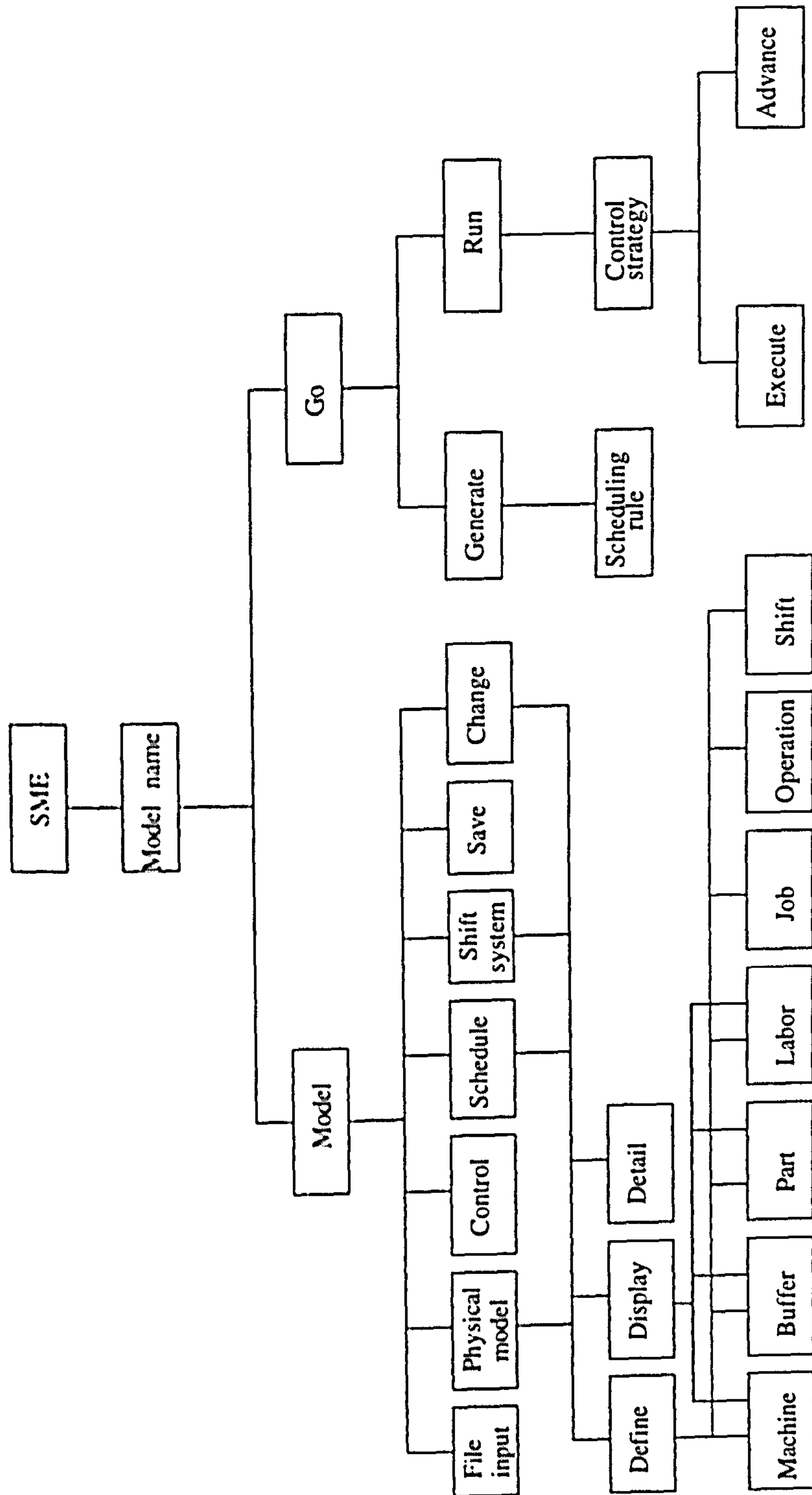


Figure 6.12: The main structure of hierarchical commands in user interface

Chapter 7

Experiments

In this chapter, the experimental framework that has been used to test the effectiveness of the proposed control strategy is presented, together with the results from the experiments that have been carried out. They test the proposed control strategy by comparing its performance with that of an ODD rule based strategy in a variety of operating environments with different levels of impact of machine breakdowns.

7.1 Objective and design of experiments

7.1.1 Important issues on design of experiments

In the literature, models used in this kind of experiments are either constructed from real systems, or hypothetical models. The models which are based on real systems are usually used for solving particular problems in the systems, and the hypothetical models usually for studying some generic type of problems of interest.

For studying the problem of maintaining shop floor stability, hypothetical shop models have been used to capture some essential characteristics of job shops.

The important elements in a job shop can be classified into three categories: system related, plan related and control related, which respectively describe resource capacity, resource requirements, and objectives and ways by which the plan would be executed.

The major system related elements in a dual-resource job shop with machine breakdowns include:

- 1) number of machines,
- 2) number of workers,
- 3) shift system,
- 4) overtime availability and pattern,
- 5) machine breakdown pattern.

The major plan related elements include:

- 1) number of jobs and operations,
- 2) processing time distribution,
- 3) flow pattern of jobs through the shop,
- 4) the shop load level,
- 5) due dates assignments.

The major control related elements include:

- 1) job release strategy,
- 2) control strategies,
- 3) overtime strategy,
- 4) performance evaluation criteria.

7.1.2 The production system

In the literature, size and structure of production system models used for experiments vary from one study to another. The number of machines in a model ranges from only a few, e.g. less than 10 in [Ben-Arieh et al 1989] [Sarin et al 1990], to several dozen or more. In machine scheduling studies, a model which consists of 9 machines is often seen as adequate to represent the complex structure of a job shop since studies, cited by [Conway 1964], have shown that increasing the number of machines from 9 to 27 had no significant effect on the relative performance of decision rules.

The hypothetical model used in this study is composed of 15 machines based on the consideration that in a DRC environment, 15 machines should be enough to represent the complexity of such a system if 9 machines is seen as adequate to represent the structure of job shops in a single resource environment. In the model, each workcentre has a single machine, which is a system setting where a machine breakdown can have the most serious effects on performance since it completely breaks the supply chain. Each workcentre has an input buffer with assumed infinitive capacity.

The system has two groups of workers, an operator group with 12 workers for processing jobs, and a setter group with 6 workers for setting up the machines; in a DRC environment, the number of workers is usually less than the number of machines, and in a real manufacturing system, jobs are usually not allowed to remain waiting for long due to lack of workers. All the workers are assumed to be equally efficient at each work centre. Each job requires both an operator and a machine, and no operator can simultaneously work on more than one machine.

A typical shift system of a job shop has been used in the study, in which each week has 5 working days, and each working day has one eight hour shift. Since in a real manufacturing environment, there are often some constraints on using

overtime, e.g. limited number of hours per day, etc, in the model, therefore, two hours are allowed for overtime at the end of each day; and all the labour in the day shift is assumed to be available for overtime as well.

Machine breakdown is one of the most commonly occurring disruptive events. According to industrial statistics reported by [Sharma 1987], average breakdown of individual machines varied between 0-40 hours per month. In the literature, different levels and frequencies of machine breakdowns are usually used to study their effects on system performance, e.g. [Sheu & Krajewsk 1994] used three levels of machine breakdown: 0, 10, 20 hours per machine per week, and [Ben-Arieh et al 1989] used two machine breakdown frequencies, average two times and four times for each simulation run in his experiments.

For testing performance of the proposed control strategy at different machine breakdown levels, experiments in this research are carried out at three levels of machine breakdown times, 2.5 hours, 5 hours and 7.5 hours per machine per week respectively. For each level of machine breakdown, three different breakdown frequencies, 2, 3 and 4 times per machine per week, have been used to generate machine breakdown models. Exponential and uniform distributions have respectively been used to model the interval of machine breakdowns and machine repair times in the experiments since these are the distributions often used in research studies, e.g. [Ben-Arieh et al 1989], [Sheu & Krajewsk 1994], and they provide an appropriate representation of machine breakdown characteristics. Table 7.1 shows the mean values used in these distributions. Minimum and maximum of the uniform distributions are respectively 0.5 and 1.5 times the mean values shown in the table.

7.1.3 The production plan

The plan used for the experiments has 50 jobs and each job is assumed to be a unique order. Several parameters of the models are randomly generated. The

Table 7.1: Mean times used in machine breakdown distributions

<i>Breakdown</i>		<i>Time Interval</i>	<i>Repair time</i>
Level	Frequency	(minutes)	(minutes)
2.5 hours (low)	2	1200	75
	3	800	50
	4	600	37.5
5 hours (medium)	2	1200	150
	3	800	100
	4	600	75
7.5 hours (high)	2	1200	225
	3	800	150
	4	600	112.5

number of operations per job varies randomly from 3 to 13 operations with a mean of about 10, and the total number of operations is about 470. Routings of jobs, i.e. machining sequences of jobs were also randomly chosen, and a job can be routed through the same machine more than once. The routings are assumed to be fixed after generation.

These jobs can be newly arriving jobs to a shop, or jobs which have not been completed in the last scheduling time period and therefore need to be rescheduled and processed with the newly arriving jobs in the next scheduling time period. But, it is assumed that there is no overdue job at the beginning of the scheduling period.

Operation time is composed of setup and processing times; in the literature, they are usually generated by three types of distributions, exponential, normal and uniform and [Dar-El & Wysk 1982] hypothesised that “real life situations will yield processing time distribution following somewhere within the range offered

by these three types". In some static shop environments, some researchers, e.g. [Dar-El & Wysk 1982], have demonstrated that different processing time distributions could favour one type of dispatching rule over others. Exponential distribution is often used in studies on static environments to describe a large range of variations between different operations. In a dynamic environment (with impact of disruptive events, e.g. machine breakdown), however, such variations would be often imposed by disruptive events, e.g. machine breakdown, and it is these disruptive events which have a more significant effect on system performance rather than the type of distribution used for generation of operation times. Therefore, this study has simply adopted the methods used by [Scudder et al 1993] [Dar-El & Wysk 1982] and [Waikar et al 1995] for generation of operation times.

In the study, setup times are generated by a uniform distribution between 10 minutes and 100 minutes. Processing times are generated by a normal distribution, with different mean value and standard deviation to produce models with different shop loads. The mean value and standard deviation (SD) were derived as follows:

$$\text{Mean} = (L - S) / N$$

where L is shop load, S is total setup times, and N is the total number of operations.

$$\text{SD} = 0.2 \text{ Mean}$$

The above method, by which the standard deviation is chosen, was used by [Waikar et al 1995].

Shop load is defined as the percentage of time that the machines are busy on average. [Dar-El & Wysk 1982] classified light, medium and heavy loading as 70%, 77%, and 85% shop loads respectively. They suggested that loads much

below 70% result in inefficient plant utilisation. In this study, therefore, three models with 70, 77 and 85 percentage of machine utilisation have been used for the experiments.

Set up times have been assumed to be fixed, i.e. independent of operation sequence. Set up times also remained the same for the three models with different shop load levels. In other words, shop loads were increased or decreased by changing processing times of operations only. Hence, changes in shop loads can be seen to result from changing batch sizes for the jobs. Total set up times in the study is 6720 minutes, and about 28% of operations need set ups. Since set up times were fixed in all the models, the utilisation of operators increased in the model with higher shop load levels.

Operator utilisation is about 60% at 70% shop load, 68% at 77% shop load, and 75% at 85% shop load, which are in the range of 50% - 75% staffing levels within which, as indicated by [Nelson 1967] and [Trelevent 1987], a dual resource constrained system can operate most efficiently. Setter utilisation in the schedule was very low, so that the setter was not a constraining resource most of the time during execution of the simulation model.

[Conway et al 1967] found that the performance of all rules related to mean lateness and number of tardy jobs were somewhat sensitive to the method of due date establishment. Due date is usually determined on the basis of either the total work content of the job (TWK) or the number of operations (NOP), allowing some slack in the process [Holloway & Nelson 1974]. Since TWK is not only the most widely used due date assignment rule in the literature [Gupta et al 1989], but also used in many industrial environments, it has been used in the generation of job due dates in this study.

By TWK, a job's due date is calculated on the basis of its total operation time, i.e. $\text{job's due date} = k * T$, in which T is total operation time (including setup time),

and k is a multiplier. [Conway et al 1967] used $k = 9$ in their research, i.e. a job could spend a total of 8 times its total operation time waiting in queues before becoming tardy. [Scudder et al 1993] set due dates by using the TWK method with multipliers of 3, 6 and 9 for providing approximately 'tight', 'moderate' and 'loose' settings for their particular simulation model. [Raman & Talbot 1993] used four levels of due date tightness with multipliers of 2, 3, 4 and 5 in their study. In a typical job shop, queuing time can be over 90% of total lead time for a job. But, due to market competition, many manufacturing companies have made a lot of effort in reduction of queuing time by a variety of means, e.g. Group Technology (GT); therefore, a multiplier of 3.5 has been used to reflect this tendency. In this study, a job's due date can be either at the end of the particular day as determined by its total work content multiplied by 3.5, or at the end of the last day of the plan if the calculated due date by this approach is over the schedule time span (which is two weeks).

From a scheduling point of view, as revealed by a research [Muhlemann et al 1982], in a manufacturing environment with impact of machine breakdowns, system performance deteriorates as the rescheduling period increases. However, if the schedule time span is too low, it may not be able to discriminate properly (without a large number of runs) the effect of different control strategies; for examples, some strategies may favour jobs with immediate due dates but at the expense of other jobs with later due dates, which then may become late as a consequence. Also if a short time span is used, it may fail to show clearly the effect of capacity compensation measures, i.e. overtime in this research. Hence, two weeks was chosen as the time span instead of one week scheduling period as is often the practice in industry.

7.1.4 Control related elements

All the jobs are released at the beginning of the scheduling period. As discussed in chapter 3, the effectiveness of an ORR strategy is much dependent upon the

planning system as well as the variance control of the shop floor. In a dynamic manufacturing environment, as many simulation studies in which exponential distribution was used for generation of operation processing times, “the most effective strategy for reducing mean tardiness and proportion tardy is to release jobs immediately to the shop floor.” [Melnik et al 1994]

Two dispatching strategies were used in the experiments for real-time control, the proposed control strategy and ODD rule. As described in previous chapters, ODD is one of most widely used priority rules in shop floor control. ODD based priority rules have also been favoured by many researchers, e.g. [Raman & Talbot 1993], [Scudder et al 1993], in studying due date related problems in scheduling and control. As observed by [Scudder et al 1993], “In practice, the use of rules which utilise operation-due-date information has shown great promise.” ODD rule was used instead of EDD because the due date settings in the hypothetical systems are ‘tight’ and as [Rohleder & Mabert 1988] found that in very tight conditions, the operation due date rules perform better than job based rules.

The overtime strategy as described in chapter 5 has been used in the experiments, i.e. overtime is used at any workcentre which is currently working on a tardy job (for which the current operation has already missed the operation due date) or any workcentre which is overloaded, or any workcentre which can operate a tardy job in the overtime (a workcentre may need to complete current operation before working on a tardy job). As a reflection of common practice, once labour is scheduled for working in overtime, 2 hours of overtime cost will be accounted for even if a particular worker has not been assigned full work load in the overtime period.

Many of the published studies on dispatching rules have used functions of flow-time, lateness, and tardiness as measures of the effectiveness of dispatching rules and strategies. These criteria may be defined as follows:

Flowtime (F_i): The amount of time job i spends in the system.

Lateness (l_i): The amount of time by which the completion time of job i exceeds its due date. Lateness may be negative, indicating an early completion.

Tardiness (T_i): The positive lateness of a job: $(T_i) = \max(0, L_i)$

Of these criteria, manufacturing companies prefer using due date related criteria, e.g. tardiness and lateness, rather than non-due date related measures such as minimising mean flowtime which, as pointed out by [Blackstone et al 1982], does not minimise mean tardiness even for a single machine in a static model. Between tardiness and lateness related criteria, the former, i.e. tardiness related criteria, e.g. mean tardiness, are seen as better measures [Blackstone et al 1982], since many manufacturing companies will certainly get to be punished by late delivery in one way or another, but can get little rewards from having jobs finished far in advance of their due dates. If overtime is to be used in minimising the effect of disruptions, then the cost of overtime also needs to be minimised.

The objective of maintaining shop floor stability is represented by three separate measures in this study, the number of tardy jobs, total tardiness and overtime manhours.

In the experiments, the data about these three performance measures are collected when all the jobs have been completed in each simulation run.

7.2 Results from the experiments

A series of experiments have been conducted to determine how the proposed control strategy affects the performance measures under manufacturing environments involving machine breakdowns. As described in the above section, the experiments have been done by a combination of the following conditions:

Shop loads - 70%, 77% and 85%,

Machine breakdown levels - 2.5, 5, and 7.5 per machine per week,

Machine breakdown frequencies - 2, 3, and 4 times per machine per week,

Dispatching strategy - ODD and proposed control strategy,

Capacity control - with overtime, or without overtime.

In summary, the experiments have been carried out on a total of 108 system settings. For each experimental setting, simulation run has been repeated with different random number streams for machine breakdown distributions. Same set of random number streams have been used for each experimental setting.

The experiments have been carried out in two stages. In the first stage, each experiment was run with two different random number streams for the 108 system settings for preliminary analysis; in the second stage, experiments were repeated on some of the system settings for detailed analysis using another six random number streams. The experiments aimed to provide quantitative information to compare performance of two control strategies rather than precise performance measures at each experimental setting.

7.2.1 Results from the preliminary experiments

Tables 7.2-7.7 shows the simulation results from the preliminary experiments. The results in Tables 7.2, 7.3 and 7.4 were obtained from the experiments where overtime was not used, and the results in Tables 7.5, 7.6 and 7.7 were obtained from the experiments in which overtime was available; in each case, an average of the two runs is presented in the tables.

In general, it can be seen from Tables 7.2, 7.3 and 7.4, system performance was sensitive to shop load as well as the levels of machine breakdowns when overtime

Table 7.2: Results from experiments at 70% shop load level without overtime

<i>Control strategy</i>	<i>Breakdown Level</i>	<i>Breakdown Frequency</i>	<i>Number of tardy job</i>	<i>Total Tardiness (minutes)</i>
ODD	Low	2	0	0
		3	0	0
		4	0	0
	Medium	2	4	1518
		3	0	1
		4	3	705
	High	2	16	4964
		3	11	3885
		4	12	2220
Proposed control strategy	Low	2	0	0
		3	0	0
		4	0	0
	Medium	2	4	1178
		3	2	146
		4	3	376
	High	2	14	5716
		3	9	3976
		4	10	2196

Table 7.3: Results from experiments at 77% shop load level without overtime

<i>Control strategy</i>	<i>Breakdown Level</i>	<i>Breakdown Frequency</i>	<i>Number of tardy job</i>	<i>Total Tardiness (minutes)</i>
<i>ODD</i>	Low	2	8	829
		3	4	704
		4	3	181
	Medium	2	16	4045
		3	6	938
		4	17	4108
	High	2	23	13910
		3	22	13586
		4	21	9218
Proposed control strategy	Low	2	5	854
		3	4	294
		4	6	730
	Medium	2	14	4499
		3	8	988
		4	12	3538
	High	2	20	12092
		3	20	9524
		4	19	8428

Table 7.4: Results from experiments at 85% shop load level without overtime

<i>Control strategy</i>	<i>Breakdown Level</i>	<i>Breakdown Frequency</i>	<i>Number of tardy job</i>	<i>Total Tardiness (minutes)</i>
ODD	Low	2	18	2598
		3	19	4231
		4	17	2427
	Medium	2	25	10398
		3	21	6046
		4	24	7521
	High	2	26	17753
		3	27	21937
		4	27	16926
Proposed control strategy	Low	2	11	2107
		3	15	2199
		4	12	1939
	Medium	2	17	6588
		3	17	3826
		4	18	5073
	High	2	22	14976
		3	24	16416
		4	24	11675

Table 7.5: Results from experiments at 70% shop load level with overtime

<i>Control strategy</i>	<i>Breakdown Level</i>	<i>Breakdown Frequency</i>	<i>Number of tardy job</i>	<i>Total Tardiness (minutes)</i>	<i>Overtime (hours)</i>
ODD	Low	2	0	0	0
		3	0	0	0
		4	0	0	0
	Medium	2	4	1024	10
		3	0	0	1
		4	1	364	8
	High	2	10	3533	30
		3	8	2686	25
		4	10	1540	25
Proposed control strategy	Low	2	0	0	0
		3	0	0	0
		4	0	0	0
	Medium	2	4	978	7
		3	0	23	2
		4	2	302	1
	High	2	8	3962	28
		3	7	3228	26
		4	6	1042	20

Table 7.6: Results from experiments at 77% shop load level with overtime

<i>Control strategy</i>	<i>Breakdown Level</i>	<i>Breakdown Frequency</i>	<i>Number of tardy job</i>	<i>Total Tardiness (minutes)</i>	<i>Overtime (hours)</i>
ODD	Low	2	0	52	15
		3	1	48	14
		4	1	14	6
	Medium	2	10	1494	32
		3	2	137	14
		4	10	1204	38
	High	2	16	6848	64
		3	16	8498	74
		4	16	3354	61
Proposed control strategy	Low	2	2	250	16
		3	2	63	6
		4	3	310	15
	Medium	2	10	1754	45
		3	4	566	18
		4	8	995	41
	High	2	14	6170	54
		3	14	6220	66
		4	13	4249	55

Table 7.7: Results from experiments at 85% shop load level with overtime

<i>Control strategy</i>	<i>Breakdown Level</i>	<i>Breakdown Frequency</i>	<i>Number of tardy job</i>	<i>Total Tardiness (minutes)</i>	<i>Overtime (hours)</i>
ODD	Low	2	3	198	51
		3	5	539	49
		4	4	302	44
	Medium	2	13	3163	79
		3	10	964	65
		4	15	2386	77
	High	2	18	8236	108
		3	21	12200	124
		4	18	4714	128
Proposed control strategy	Low	2	4	640	36
		3	6	948	48
		4	4	809	33
	Medium	2	10	2542	56
		3	8	1404	56
		4	14	2954	71
	High	2	18	8413	97
		3	20	10355	93
		4	18	5770	107

has not been used; system performance deteriorates with increase in shop loads and/or average machine breakdown time (per machine per week).

As shown in Tables 7.2, 7.3 and 7.4 (the experimental settings without overtime), the proposed control strategy yielded better performance (both on the number of tardy jobs and total tardiness) than the ODD rule in the hostile environments (high shop load, high level of machine breakdown). In system settings with 85% shop load, as shown in Table 7.4, and the system settings with 77% loads but having high intensity of machine breakdowns, the proposed control strategy gave better performance than the ODD rule in every experiment. This indicates that the control strategy is more robust to the impact of the machine breakdowns than the ODD rule in the hostile environments. In other situations, neither the proposed control strategy nor the ODD rule has shown superior performance; in some of these cases, the proposed control strategy performs better, and in some of the others the ODD rule resulted in better system performance than the proposed control strategy.

When overtime has been used, as shown in Tables 7.5, 7.6 and 7.7, system performance in all production settings (no matter which dispatching strategy has been used) have generally improved compared with the results in Table 7.2, 7.3 and 7.4, except the loosest ones where system could meet due date requirements without overtime. Obviously, such improvement was achieved at the expense of overtime cost, and the highest overtime cost incurred was 128 manhours. The same overtime control strategy, as introduced in chapter 5, has been used.

As shown in Tables 7.5, 7.6 and 7.7, the proposed control strategy seems to perform better than the ODD rule in hostile environments with heavily loaded shop and/or high intensity of machine breakdown, especially on overtime hour measure. This can be seen from results derived from the situations with 85% of shop load and, medium and high level of machine breakdowns, and the situation with 77% shop load and high intensity of machine breakdowns. In these cases,

either the proposed control strategy outperformed the ODD rule on all three measures, or on two of the three measures. In some of the cases, the ODD rule resulted in lower total tardiness, but in every case, such saving on tardiness was at the expense of higher overtime hours and possible poorer results on number of tardy jobs.

In other situations, especially in the system settings with 70% shop load, and those with 77% shop load and, low and medium intensity of machine breakdowns, the results have not shown a consistent pattern to indicate a superior dispatching strategy. In some situations, the proposed control strategy performed better, and in some others, the ODD has shown better performance; in other situations, the results are mixed and it is difficult to compare the two control strategies since no assumption on cost structure were made in this research. But, as pointed out above, in many cases as shown in Table 7.6 and 7.7, improvements in total tardiness by the ODD rule were at the expense of poorer performance on number of tardy jobs and/or overtime costs. In some cases, e.g. for 85% shop load with high level of machine breakdowns and breakdown frequency of 2 times per week, the ODD rule produced only narrowly better results on total tardiness but with much poorer performance on overtime manhours. In these cases, saving in reduction of total tardiness (minutes) by the ODD rule may not be able to offset the increase in overtime labour hours.

To confirm the findings from the preliminary experiments, the simulation runs were repeated with 6 further random number streams for system settings with 77% and 85% of shop loads to obtain the data necessary for statistical analysis.

7.2.2 Results from detailed analyses

Detailed analysis was carried out on the system settings with 77% and 85% of shop loads, and based on the data collected from a total of 8 simulation runs for each experiment (including the two runs in the preliminary experiments).

Mean values of the performance measures are shown in Tables 7.8, 7.9, 7.10 and 7.11. In general, these results have confirmed the critical conclusions derived from the preliminary experiments, i.e. when overtime is not available, as shown in Tables 7.8 and 7.9, the proposed control strategy is more robust than the ODD rule under impact of machine breakdown in hostile environments; when overtime has been used, as shown in Tables 7.10 and 7.11, it gave better performance on the number of tardy jobs with lower overtime hours in hostile environments.

The results have also shown that the frequency of machine breakdowns has some influence on performances of the proposed control strategy in some system settings, especially for the situations when overtime has not been used. As shown in tables 7.8 and 7.9, the proposed control strategy performed better in the experimental settings with low frequency of machine breakdowns.

A paired t-test has been used for further confirmation of this analysis. For the paired t-test, data was collected in pair under homogeneous conditions, i.e. same shop load, same distribution parameters for machine breakdown interval and repair time, etc, but different dispatching strategies. Suppose that $(X_{11}, X_{21}), (X_{12}, X_{22}), \dots, (X_{1n}, X_{2n})$ are a set of n observations derived by using ODD rule and the proposed control strategy, and $X_1 \sim N(\mu_1, \sigma_1^2)$ and $X_2 \sim N(\mu_2, \sigma_2^2)$, then the differences between each pair of observations $D_j = X_{1j} - X_{2j}$, is normally distributed with means:

$$\mu_D = E(X_1 - X_2) = \mu_1 - \mu_2$$

In order to compare the effectiveness of the control strategy against the ODD rule for the measures of mean number of tardy jobs and mean tardiness, the following Null and alternative hypothesis were set up:

$$H_0 : \mu_D = 0$$

Table 7.8: Mean values of results at 77% shop load level without overtime

<i>Control strategy</i>	<i>Breakdown Level</i>	<i>Breakdown Frequency</i>	<i>Number of tardy job</i>	<i>Total Tardiness (minutes)</i>
ODD	Low	2	3.1	422
		3	2.8	266
		4	3.4	238.3
	Medium	2	16.5	4624
		3	12.4	2356
		4	13.9	2879
	High	2	22.3	12877
		3	22.0	10562
		4	20.8	7831
Proposed control strategy	Low	2	3.9	308
		3	2.9	268
		4	3.9	439
	Medium	2	13.6	3897
		3	12.9	2909
		4	13.0	3709
	High	2	18.9	10179
		3	19.0	8266
		4	19.6	8088

Table 7.9: Mean values at 85% shop load level without overtime

<i>Control strategy</i>	<i>Breakdown Level</i>	<i>Breakdown Frequency</i>	<i>Number of tardy job</i>	<i>Total Tardiness (minutes)</i>
ODD	Low	2	17.5	3316
		3	17.9	3244
		4	16.9	3542
	Medium	2	22.8	9464
		3	21.8	7669
		4	23.0	7411
	High	2	27.0	18681
		3	26.9	17803
		4	27	17808
Proposed control strategy	Low	2	11.9	2341
		3	11.8	1869
		4	11.9	2373
	Medium	2	16.8	6789
		3	19.1	6360
		4	18.1	5544
	High	2	24.0	15603
		3	24.1	13780
		4	24.8	14271

Table 7.10: Mean values of results at 77% shop load level with overtime

<i>Control strategy</i>	<i>Breakdown Level</i>	<i>Breakdown Frequency</i>	<i>Number of tardy job</i>	<i>Total Tardiness (minutes)</i>	<i>Overtime (hours)</i>
ODD	Low	2	0.5	27	8.5
		3	0.4	13	7.8
		4	0.8	85	7.0
	Medium	2	9.8	1665	39.0
		3	5.9	993	31.3
		4	6.9	826	34.3
	High	2	16.3	6921	69.0
		3	15.4	5990	68.0
		4	13.8	3083	53.5
Proposed control strategy	Low	2	1.0	139	9.5
		3	1.0	51	6.8
		4	1.6	165	11
	Medium	2	8.0	2485	33.8
		3	7.9	1738	32.0
		4	7.6	1660	34.3
	High	2	13.5	6164	53.8
		3	13.1	4825	60.0
		4	13.8	4818	56.8

Table 7.11: Mean values of results at 85% shop load level with overtime

<i>Control strategy</i>	<i>Breakdown Level</i>	<i>Breakdown Frequency</i>	<i>Number of tardy job</i>	<i>Total Tardiness (minutes)</i>	<i>Overtime (hours)</i>
ODD	Low	2	3.3	346	51
		3	4.9	449	52
		4	4.7	440	49
	Medium	2	13	3260	65
		3	12.3	2205	72
		4	14.4	2453	79
	High	2	19.8	8959	109.5
		3	20.3	8770	108.5
		4	20	6176	116
Proposed control strategy	Low	2	3.8	553	35
		3	4.9	673	36
		4	4.6	747	38
	Medium	2	11.1	2982	53.3
		3	11	2677	67
		4	12.4	2864	68
	High	2	18.8	8364	98.8
		3	17.6	8135	96.8
		4	16.5	6203	99.8

$$H_1 : \mu_D \neq 0$$

Where μ_D = mean of the pairwise difference for the particular performance measure of interest.

The following equation [Hines & Montgomery 1990] has been used for the tests:

$$t_0 = \frac{\bar{D}}{S_D/\sqrt{n}}$$

Where

$$\bar{D} = \frac{\sum_{j=1}^n D_j}{n}$$

and

$$S_D^2 = \frac{\sum_{j=1}^n D_j^2 - [(\sum_{j=1}^n D_j)^2/n]}{n-1}$$

A significant positive t_0 would imply that the control strategy performed significantly better than the ODD rule for the particular measure; a significant negative t_0 would imply that the ODD rule performed better.

The test has been carried out in two different ways. One is for testing the hypothesis at system settings with the same shop load and same breakdown distribution parameters; for each test, 8 pairs of data were available for analysis. In the other case, the hypothesis has been tested at the same machine breakdown intensity level (three levels, low, medium and high; each level includes three breakdown frequencies), and 24 pairs of data were available for each test. The results of the paired t-tests are presented in Tables 7.12, 7.13, 7.14 and 7.15, in which:

- confidence level, e.g. 95%, means that t_0 is significantly positive, and the control strategy performs better than the ODD rule at this confidence level,

- confidence level (O), e.g. 95% (O), means that t_0 is significantly negative on this measure, and the ODD rule gave better performance than the control strategy with this level of confidence;
- NS: t_0 is Not Significant at 95% and neither control strategy gave significantly better performance.

Overtime was presented by the mean value of differences derived from the two control strategies.

On the whole, these results have confirmed the findings from the preliminary analysis, i.e. the proposed control strategy outperforms the ODD rule in hostile environments, and in other situations, performance of the two control strategies are not significantly different. When overtime has been used, in some cases, the ODD rule gave better performance on total tardiness but at the expense of higher overtime.

When overtime was not used, for experimental settings with 85% shop load (as shown in Table 7.14), the control strategy gave significantly better results in all but three cases with greater than 95% confidence; for the settings with 77% shop load (as shown in Table 7.12), the null hypothesis has been rejected (in favour of the control strategy) in half of the cases at high intensity machine breakdown level. This can also be seen from the results in Tables 7.13 and 7.15, in which for all the system settings with 85% shop load, and the system setting with 77% shop load and high intensity of machine breakdown, the null hypotheses have all been rejected (in favour of the control strategy) with greater than 95% confidence. When overtime has been used, for these experimental settings, most t-test results on total tardiness are NS - not significant; all the results on the number of tardy jobs are either NS, or significant positive; all the results but one on overtime measure indicated that the control strategy used less overtime hours than the ODD rule. As shown in 7.12, in only one 'hostile' situation (77% shop load, high

Table 7.12: t-test (paired) at 77% shop load level

<i>Breakdown Level</i>	<i>Breakdown Frequency</i>	<i>Number of tardy job</i>	<i>Total Tardiness</i>	<i>Diff. in overtime hours used (ODD - control strategy)</i>
Low	2	NS	NS	No
	3	NS	NS	No
	4	NS	NS	No
Medium	2	99.5%	NS	No
	3	NS	NS	No
	4	NS	NS	No
High	2	99.8%	99.9%	No
	3	99.8%	NS	No
	4	NS	NS	No
Low	2	NS	NS	-1
	3	99.5% (O)	NS	1
	4	NS	NS	-4
Medium	2	NS	NS	5.3
	3	NS	99.5% (O)	-0.8
	4	NS	99% (O)	0.0
High	2	98%	NS	15.3
	3	NS	NS	8.5
	4	NS	98% (O)	-3.3

Table 7.13: t-test (paired) at 77% shop load level for different breakdown levels

	<i>Breakdown Level</i>	<i>Number of tardy job</i>	<i>Total Tardiness</i>	<i>Diff. in overtime hours used (ODD - control strategy)</i>
No overtime	Low	NS	NS	
	Medium	NS	NS	
	High	99.9%	98%	
With overtime	Low	98% (O)	NS	-0.8
	Medium	NS	99.9% (O)	1.5
	High	95%	NS	6.8

intensity machine breakdown with breakdown frequency of 4 times per machine per week), the ODD rule performed better than the control strategy. But when all the performances at the same machine breakdown level were taken into account together, as shown in Table 7.13, the control strategy generally outperformed the ODD rule at this machine breakdown level.

For ‘non-hostile’ situations, when overtime has not been used, the control strategy outperformed the ODD rule for the system settings with 85% shop load; and for most of other cases (i.e. system settings with 77% shop load and low or medium intensity of machine breakdowns), the results were not significant enough for supporting either of the two control strategies. When overtime has been used, for the system settings with low intensity machine breakdowns at 85% shop load, the ODD rule gave better performance on total tardiness at this level, as shown in Table 7.15, but was at the expense of higher overtime. In the experimental settings with 77% shop load and low or medium intensity of machine breakdowns, the ODD rule seems to have performed better than the control strategy.

If the two control strategies are compared with each other according to comparable measures, the following conclusions can be reached:

- When overtime has not been used, the proposed control strategy performs

Table 7.14: t-test (paired) at 85% shop load level

<i>Breakdown Level</i>	<i>Breakdown Frequency</i>	<i>Number of tardy job</i>	<i>Total Tardiness</i>	<i>Diff. in overtime hours used (ODD - control strategy)</i>
Low	2	99.9%	95%	No
	3	99.5%	99.5%	No
	4	99.5%	NS	No
Medium	2	99.9%	99.5%	No
	3	99%	NS	No
	4	99.5%	99%	No
High	2	99%	99%	No
	3	NS	99.5%	No
	4	98%	99.5%	No
Low	2	NS	NS	16
	3	NS	NS	15
	4	NS	NS	11.3
Medium	2	99.5%	NS	16.5
	3	NS	NS	3.3
	4	NS	NS	11.3
High	2	NS	NS	10.8
	3	98%	NS	11.8
	4	95%	NS	16.5

Table 7.15: t-test (paired) at 85% shop load level for different breakdown levels

	<i>Breakdown</i>	<i>Number of</i>	<i>Total</i>	<i>Diff. in overtime hours used</i>
	<i>Level</i>	<i>tardy job</i>	<i>Tardiness</i>	<i>(ODD - control strategy)</i>
No overtime	Low	99.9%	99.9%	
	Medium	99.9%	99.9%	
	High	99.9%	99.9%	
With overtime	Low	NS	95% (O)	14.0
	Medium	99.5%	NS	10.3
	High	99.9%	NS	13.0

better than the ODD rule in hostile environments; in other cases, the results are not significant.

- When overtime was used, the proposed control strategy still performs better than the ODD rule in hostile environments. In other cases, the ODD rule gave better performance at two levels with 77% shop load: low level of machine breakdown for number of tardy jobs, and medium level of machine breakdown for total tardiness. On system settings with 85% shop load and low machine breakdown, the results are mixed since saving on total tardiness by the ODD rule is at the expense of poorer performance on overtime hours.

Tables 7.16 and 7.17 provide the mean value and 95% confidence interval on the difference in performance for the experimental settings at 77% and 85% shop load.

7.2.3 Discussion of results

In general, performance of the proposed control strategy is situation related. It performs better in hostile situations which have been described as heavily loaded shop and high intensity level of machine breakdowns in this study. In

Table 7.16: Mean and confidence interval for 77% shop load level (ODD - control strategy)

<i>Breakdown</i>		<i>Number of tardy job</i>			<i>Total Tardiness</i> <i>(minutes)</i>			<i>Overtime</i> <i>hours used</i>
<i>Level</i>	<i>Freq.</i>	<i>Mean diff.</i>	<i>95% Conf.</i>		<i>Mean diff.</i>	<i>95% Conf.</i>		<i>Mean diff.</i>
Low	2	0.75	-1.38	2.88	-115	-308	79	No
	3	-0.13	-1.42	1.17	-1	-301	299	No
	4	-0.5	-3.21	2.21	-201	-599	198	No
Medium	2	2.88	1.24	4.51	727	-438	1892	No
	3	-0.5	-3.03	2.03	-553	-1269	163	No
	4	0.88	-3.03	4.78	-830	-1764	104	No
High	2	3.38	1.90	4.85	2698	964	4431	No
	3	3.0	1.59	4.41	2297	-892	5485	No
	4	1.13	-1.95	4.20	-256	-1690	1177	No
Low	2	-0.5	-1.39	0.39	-111	-285	63	-1.0
	3	-0.63	-1.25	-0.003	-38	-79	2	1.0
	4	-0.88	-2.09	0.34	-81	-338	176	-4.0
Medium	2	1.75	-0.47	3.97	-820	-1674	34	5.25
	3	-2.0	-4.28	0.28	-746	-1126	-366	-0.75
	4	-0.75	-2.22	0.72	-834	-1383	-285	0.0
High	2	2.75	0.81	4.69	757	-1218	2733	15.25
	3	2.25	-1.26	5.76	1165	-1158	3488	8.5
	4	0.0	-2.61	2.61	-1735	-3018	-452	-3.25

Table 7.17: Mean and confidence interval for 85% shop load level (ODD - control strategy)

<i>Breakdown</i>		<i>Number of tardy job</i>			<i>Total Tardiness</i> <i>(minutes)</i>			<i>Overtime</i> <i>hours used</i>
<i>Level</i>	<i>Freq.</i>	<i>Mean diff.</i>	<i>95% Conf.</i>		<i>Mean diff.</i>	<i>95% Conf.</i>		<i>Mean diff.</i>
Low	2	5.63	3.79	7.46	975	191	1759	No
	3	6.13	2.59	9.66	1374	618	2130	No
	4	5.0	2.21	7.79	681	-765	2127	No
Medium	2	6.0	3.47	8.53	2675	1345	4004	No
	3	2.63	0.96	4.29	1309	-108	2726	No
	4	4.88	2.21	7.54	1867	663	3070	No
High	2	3.0	1.16	4.84	3078	1089	5067	No
	3	2.75	-0.03	5.53	4023	1967	6078	No
	4	2.25	0.59	3.91	3537	1705	5368	No
Low	2	-0.5	-2.60	1.60	-207	-492	78	16.25
	3	0.0	-2.05	2.05	-224	-699	251	15.0
	4	0.13	-3.02	3.27	-307	-847	235	11.0
Medium	2	2.38	1.04	3.71	153	-467	773	16.5
	3	1.25	-1.84	4.34	-472	-1099	155	3.25
	4	2.0	-0.32	4.32	-411	-1213	391	11.25
High	2	1.0	-1.0	3.0	595	-2040	3230	10.75
	3	2.63	0.63	4.62	635	-516	1786	11.75
	4	3.5	0.47	6.53	-27	-1200	1145	16.5

these situations, the proposed control strategy clearly produced better system performance than the ODD rule in most of the system settings when no overtime was available. This is because the proposed control strategy could use machine related information, e.g. machine breakdown and current loadings, in dispatching decisions to prevent or ease capacity problems caused by the disruption and improve production flow. In contrast, the ODD rule, which does not use such information, was more likely to generate congested routes. In relatively benign environments, however, the results are mixed; there is no consistent pattern or significant results to identify a superior control strategy. It demonstrates that a simple priority rule can perform well in relatively loose conditions. As noticed by [Dar-El & Wysk 1982], "At the lighter shop load level, most priority rules perform well and it is difficult to discriminate between them." [Fredendall & Melnyk 1995] also observed that "when shop load ratios are low, the shop is able to handle the traffic using any priority rule."

When overtime was used, the problem of catching up tardy jobs, overloaded workcentres and congestions could be alleviated by periodical capacity compensation. In some situations, especially some cases in benign and medium shop load environments, the ODD rule could yield better performance than the proposed control strategy. In hostile environments, however, two hours of overtime a day might not be enough to ease the capacity problems to the level at which the dispatching mechanism could perform well without knowledge of machine status in the shop and, therefore, more likely to give inferior performance to the proposed control strategy.

The conditions for which the proposed control strategy does not have satisfactory answers can be divided into two groups. The first case is when the system is reaching conditions for which the proposed control strategy has not been designed, e.g. there may be some situations which need to be further examined and explicitly described by the rules modelling the proposed control strategy. The second case is when the proposed control strategy does not have accurate infor-

mation for decision making, especially information on future shop status provided by the look-ahead mechanism.

As described in chapter 5, it is assumed by the Look-ahead mechanism at the time of dispatching that all the broken down machines were remaining inoperational within its look-ahead time horizon. But some of the broken down machines might recover shortly after the last dispatching action, and jobs engaged on these machines may affect and/or be affected by a selected job. In some situations, performance of the proposed control strategy is sensitive to frequency of machine breakdowns, and performed better in the experimental settings with low frequency of machine breakdowns. This may be also due to the assumption made for the Look-ahead mechanism, i.e. in system settings with low frequency of machine breakdowns, machine repair times are more likely to be longer than the look-ahead time span, so that information provided by the look-ahead mechanism will be more accurate, and the control could be more effective. But, when overtime was used, the results have not shown similar pattern. It is probably because the Look-ahead mechanism can only provide information about future shop status which does not take into account effects of overtime. In other words, use of overtime would affect accuracy of the Look-ahead mechanism.

In summary, the proposed control strategy outperformed the ODD rule in most tight system settings when overtime was not used. When overtime was used, the control strategy gave promising results for reduction of overtime cost and particularly performed better than the ODD in hostile environments.

Chapter 8

Summary and conclusions

8.1 Summary

This study has investigated real-time shop floor control strategies for dealing with contingency on the shop floor. The intent was to include both order related and machine related information in the dispatching process to improve effectiveness of shop floor control in manufacturing environments with dual resource (machine and labour) under impact of machine breakdowns.

A multiperspective dispatching strategy has been proposed for promoting shop floor stability. In this control strategy, a heuristic process is used to identify the constraints which could have the most significant impact on system performance at the time of dispatching.

Three order related factors: criticality of jobs, consequences of job selections, and predicted waiting time on succeeding workcentre were used in the dispatching mechanism. A timing plan in a predictive schedule is used to provide intermediate objectives for detecting extent of deviation of production from its original plan. The organisational goals, i.e. maintaining shop floor stability, has been measured by total tardiness, number of tardy jobs, and overtime cost.

Constraints of resource availability have been described by information (including status, work loadings, remaining work, etc) on both the dispatching workcentre and on succeeding workcentres. Levels of the constraints are described by status of the resources, e.g. broken down workcentres, and loading related measurements such as long queue workcentres, etc.

A look-ahead mechanism has been used to project current shop status, and obtain and translate future shop states in the control strategy. Through the look-ahead mechanism, information on direct upstream and succeeding workcentres were included in the dispatching process, in addition to information on candidate jobs and dispatching workcentres.

An heuristic procedure has been developed for classification of shop floor situations and coordination of these factors. A group of control policies, e.g. keep bottleneck busy, have been used in the control strategy to allow the dispatching mechanism to make decisions based on a 'broad' view on shop floor status (order perspective and resource perspective, current shop status and future shop status).

An overtime based capacity control strategy is also suggested in the study, in which both tardy jobs and overloaded workcentres have been taken into account.

For testing and analysing the control strategy, a simulated manufacturing environment has been developed, which consists of a schedule generator for predictive scheduling; a control module for modelling dispatching strategy and overtime based capacity control policy; a shop floor emulator for simulation of the production process on the shop floor and generating machine breakdown events according to given distributions about these events; a monitor for data collection and translation for use by the control module; and a user interface to model the system and conduct experiments, and output results from experiments.

A knowledge based approach has been used for modelling the proposed control

strategy because of its ability in representation of experiential knowledge of shop floor personnel, and its structure which allows easy modification of the control strategies without affecting the overall structure of the program.

The simulated manufacturing environment has been based on Knowledge Craft, an expert system development tool. An object oriented approach has been used for modelling and programming. The proposed control strategy is represented by CRL-OPS rules.

A series of experiments have been conducted in the environment for comparing performance of the proposed control strategy with a commonly used priority rule, Earliest Operation Due Date (ODD). A hypothetical manufacturing system model has been used in the experiments. The experiments were carried out at different system settings (different machine load levels, different intensity level of machine breakdowns, and with overtime and without overtime)

The experiments have been carried out in two stages, which are respectively for preliminary and detailed analysis. In the first stage, the experiments were carried out on all the system settings described above, and simulation runs have been repeated with 2 different random number streams for machine breakdown distributions. Mean values of the measurements have been used for performance comparison between the two control strategies. In the second stage, experiments have been repeated on some selected system settings (with 77% and 85% shop loads) with 6 more different random number streams for machine breakdown distributions for detailed analyses. In this stage, data was collected and presented not only independently, but also in pairs, i.e. statistical analysis was carried out based on the differences between two groups of observations on system performance obtained using different control strategies.

The experiments on a variety of shop floor conditions show:

- Performance of the proposed control strategy is situation related; it gave

better performance than the ODD rule in hostile environments, which have been described as high shop load and/or high intensity of machine breakdowns in the experimental settings; in other experimental settings, neither the proposed control strategy nor the ODD rule has shown superior performance in a consistent or significant manner.

- in the shop conditions in which there is no real-time capacity control facility to increase available resource capacity, the proposed control strategy clearly outperformed the ODD rule on delivery performance in hostile environments, and therefore demonstrated its ability to reduce impact of machine breakdowns on maintaining shop floor stability;
- in the shop conditions in which overtime is available, the proposed control strategy gave promising results in reduction of overtime cost in 'hostile' situations.

8.2 Conclusions

Following conclusions can be summarised from the study:

- The multiperspective dispatching strategy has shown great promise for improvement of effectiveness of shop floor control in dual-resource constrained manufacturing environments subject to machine breakdowns. But effectiveness of this control strategy is shown to be dependent on the environment conditions, such as shop load and intensity level of machine breakdowns. In general, it is much more effective than the ODD rule in hostile environments, whereas in benign environments, there is no significant difference in performance.
- Results of this study have supported the suggestion made by [Melnik 1988], that there is a need to modify the formal dispatching procedure in a manufacturing system for improving effectiveness of shop floor control.

- An implication of these results is that for maintaining shop floor stability, we should think more of ‘broadening’ the view of the dispatching mechanism to allow a constant smoothing of workflow, rather than use a straightforward priority rule based dispatching plus ‘firefighting’ style of schedule revision.
- The study has also shown that a knowledge based approach provides an appropriate mechanism for modelling and manipulating heuristic knowledge, and provides a highly flexible software environment which can be adapted to real problems. The flexibility provides the potential for the multiperspective control strategy (possibly with suitable enhancements) to be generally used in manufacturing environments which have other disruptive events in addition to machine breakdowns.
- Development of the simulated manufacturing environment provided an appropriate tool for investigation of control strategies. In addition to the knowledge based approach, object oriented programming approach and graphical user interfaces have also demonstrated their merits. It shows that it is crucial to develop an appropriate software tool as a platform for shop floor control related research.

8.3 Further research

The following directions for further research can be suggested:

1. The approach of coordinating multiperspectives in dispatching can be expanded by including control strategies for dealing with other random events, such as rush jobs and labour absence, in addition to machine breakdowns. The simulated manufacturing environment provides a platform on which modifications to the control strategy can be easily made.
2. Another direction of research, as suggested by [Melnik et al 1994], is to investigate the interaction between various control mechanisms: ORR, dispatching, capacity control, etc. Such research aims to uncover and understand

the effects of a combination of various control policies and different control functions in different system settings, such as different tightness of due date settings.

3. Application related research. For application of the proposed control strategy, further revision of the control strategy may be required. Since the information used in the proposed control strategy may not be available in an industrial manufacturing environment, the control strategy will need to be refined to be suitable for specific manufacturing settings. It may be possible to interface the simulated manufacturing environment to a real manufacturing system (with the latter replacing the emulator), and allow the control module to make dispatching decisions according to information provided by the shop floor monitoring and data collection system. This facilitates the migration path from shop floor control simulation to implementation.

Reference

- [Adiga & Glassey 1991] Adiga, S. and Glassey, C. R. 1991. *Object-oriented simulation to support research in manufacturing systems*. International Journal of Production Research, Vol. 29, No.12, 2529-2542.
- [Baker 1984] Baker, K. R. 1984. *Sequencing rules and due date assignments in a job shop*. Management Science, Vol. 30, 1093-1104.
- [Baptiste & Fabrel 1993] Baptiste, P and Fabrel, J. 1993. *Taking into account the rescheduling problem during the scheduling phase*. Production Planning and Control, Vol. 4, No. 3, 349-360.
- [Ben-Arieh 1986] Ben-Arieh. 1986. *A knowledge based system for simulation and control of CIM*. Proceedings of the 2nd international conference on simulation in manufacturing, Chicago 1986, IFS, 13-22.
- [Ben-Arieh et al 1989] Ben-Arieh, D., Moodie, C. L. and Chu, C. C. 1989. *Knowledge-based scheduling under unpredicted conditions: two approaches*. International Journal of Production Research, Vol. 27, No. 5, 869-882.
- [Bensama & Dubois 1988] Bensama, G. B. and Dubois, D. 1988. *OPAL: A multi-knowledge-based system for industrial job-shop scheduling* International Journal of Production Research, Vol. 26, No. 5 795-819.
- [Bertrand & Wortmann 1981] Bertrand, J. W. M. and Wortmann, J. C. 1981. *Production control and information systems for component - manufacturing shops*. Elsevier Scientific Publishing Company.

- [Bharadwaj et al 1994] Bharadwaj, A., Vinze, A. S. and Sen A. 1994. *Black-board architecture for reactive scheduling*. Expert systems with applications, Vol.1, No. 2, 55-65.
- [Bhattachayya et al 1991] Bhattachayya, S. K., Roy, R. and Huang, Y. S. 1991. *Integrated Intelligent Simulation Environment for Manufacturing Scheduling*. International AMSE conference.
- [Blackstone et al 1982] Blackstone, J. H., Phillips, D. T. and Hogg, G. L. 1982. *A state-of-the-art survey of dispatching rules for manufacturing job shop operations*. International Journal of Production Research, Vol. 20, No. 1, 27-45.
- [Browne et al 1988] Browne, J., Harhen, J. and Shivnan, J. 1988. *Production management systems - A CIM Perspective*. Addison-Wesely.
- [Browne 1988] Browne, J. 1988. *Production activity control - a key aspect of production control*. International Journal of Production Research, Vol. 26, No. 3, 415-427.
- [Buzacott 1982] Buzacott, J. A. 1982. *Optimal operating rules for automated manufacturing systems*. IEEE Transactions on Automated Control, Vol. AC-27, No. 1, 80-86.
- [Burbidge 1990] Burbidge, J. L. 1990. *Production Flow Analysis*. Oxford University Press.
- [Burke & Prosser 1989] Burke, P. and Prosser, P. 1989. *A distributed Asynchronous System for predictive and Reactive Scheduling* Technical Report AISL-42-89, Department of Computer Science, University of Strathclyde.
- [Carnegie 1988] Carnegie Group Inc, 1988. *Knowledge Craft Manual*. Pittsburgh, PA.
- [Carrie 1992] Carrie, A. 1992. *Simulation of Manufacturing Systems*. John Wiley.

- [Charalambous & Hindi 1991] Charalambous, O. and Hindi, K. S., 1991. *Review of artificial intelligence-based job-shop scheduling systems*. Information and Decision Technologies, Vol. 17, No. 3, 189-202.
- [Collinot 1989] Collinot, A. 1989. *Revising the BB1 basic control loop to control the behavior of knowledge sources*. Blackboard Architectures and Applications 27-44, New York: Academic Press.
- [Conway 1964] Conway, R. W., 1964. *An Experimental Investigation of Priority Assignment in a Job Shop*. Rond Corporation-Memorandum RM-3789-PR.
- [Conway et al 1967] Conway, R. W., Maxwell, W. L. and Miller, L. W. 1967. *Theory of scheduling*. Addison-Wesley Publishing Company.
- [Dar-El & Wysk 1982] Dar-El, E. M. and Wysk, R. A. 1982. *Job shop scheduling - a systematic approach*. Journal of Manufacturing Systems, Vol. 1, No. 1, 77-88.
- [Dutta 1990] Dutta, A. 1990. *Reacting to Scheduling Exceptions in FMS Environments*. IIE Transactions, Vol. 22, No. 4, 300-314.
- [Elleby et al 1988] Elleby, P., Farther, H. E. and Addis, T. R. 1988. *A constraint-based scheduling system for VLSI Wafer Fabrication*. Knowledge based group, Department of Computer Science, University of Reading.
- [Farhoodi 1990] Farhoodi, F. 1990. *A knowledge-based approach to dynamic job-shop scheduling*. International Journal of Computer Integrated Manufacturing, Vol. 3, No. 2, 84-95.
- [Finch & Gibert 1986] Finch, B. and Gibert, J. 1986. *Developing maintenance craft labor efficiency through an integrated planning and control system: a prescriptive model*. Journal of Operations Management, Vol. 6, No. 4, 449-459.
- [Fisher 1996] Fisher, D. 1996. *MRP at 'lightspeed'*. IIE Solutions, Vol. 28, No. 5, 50-52.

- [Fisher & Thompson 1963] Fisher, H. and Thompson, G. L. 1963. *Probabilistic learning combinations of local job shop scheduling rules*. Industrial Scheduling, Prentice-hall, 225-251.
- [Fox 1983] Fox, M. S. 1983. *Constraint-Directed Search: A Case Study of Job Shop Scheduling*. Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA.
- [Fox & Smith 1984] Fox, M. S., Smith, S. F. 1984. *ISIS-A knowledge-based system for factory scheduling*. Expert Systems, Vol. 1, No. 1, 25-49.
- [Fredendall & Melnyk 1995] Fredendall, L. D. and Melnyk, S. A. 1995. *Assessing the impact of reducing demand variance through improved planning on the performance of a dual resource constrained job shop*. International Journal of Production Research, Vol. 33, No. 6, 1521-1534.
- [Gallagher & Knight 1973] Gallagher, C. C. and Knight, W. A. 1973. *Group Technology*. London: Butterworths.
- [Ganguly 1985] Ganguly, S. 1985. *Factory scheduling - Link to automation*. Proc. Autofact, 17.1-17.5.
- [Gere 1966] Gere, W. S. 1966. *Heuristics in job shop*. Management Science, Vol. 13, 167-190.
- [Gershwin et al 1986] Gershwin, S. B., Hildebrant, R. R., Suri, R. and Mitter, S. K. 1986. *A control perspective on recent trends in manufacturing systems*. IEEE Control System Magazine, April, 3-15.
- [Goldratt 1980] Goldratt, E. 1980. *Optimized production timetables: a revolutionary program for industry*. APICS 23rd Annual International Conference Proceedings, 172-176.
- [Goldratt & Cox 1984] Goldratt, E. M, and Cox, J. 1984. *The goal: Excellence in Manufacturing*. Croton-on-Hudson, NY: North River Press, Inc.
- [Graves 1981] Graves, S. G. 1981. *A review of production scheduling*. Operations Research, Vol. 29, 646-675.

- [Gupta et al 1989] Gupta, Y. P., Gupta, M. C. and Bector, C. R. 1989. *A review of scheduling rules in flexible manufacturing systems*. International Journal of Computer Integrated Manufacturing, Vol. 2, No. 6, 356-377.
- [He et al 1993] He, Z., Yang, T. and Deal, D. E. 1993. *A multiple-pass heuristic rule for job shop scheduling with due dates*. International Journal of Production Research, Vol. 31, No. 11, 1677-2692.
- [Holloway & Nelson 1974] Holloway, C. A. and Nelson, R. T., 1974. *Job Shop Scheduling with Due Dates and Variable Processing Times*. Management Science, Vol. 20, 1264-1275.
- [Huang et al 1990] Huang, Y. G., Kanak, N. L. and Tripathi, S. K. 1990. *Reactive scheduling for a single machine: problem definition, analysis, and heuristic solution*. International Journal of Computer Integrated Manufacturing, Vol. 3, No. 1, 6-12.
- [Henning et al 1994] Henning, G. and Cerda, J. 1994. *ESA: expert scheduling assistant. A task-oriented knowledge-based expert system*. Applications of Artificial Intelligence in Engineering, 189-196.
- [Hines & Montgomery 1990] Hines, W. W. and Montgomery, D. C., 1990, *Probability and Statistics in Engineering and Management Science*. John Wiley & Sons.
- [IBM Chapter 6] IBM Chapter 6: Manufacturing Activity Planning, 1972. Communications Oriented Production Information and Control systems, Vol. 5, White Plains, N.Y.: IBM Technical Publications Department.
- [Istel 1987a] Istel, 1987. *SEE WHY user manual*.
- [Istel 1987b] Istel, 1987. *WITMESS user manual*
- [Intellicorp 1985] , Intellicorp 1985, *The Simkit software package, Knowledge-based simulation in Kee*.
- [Jackson 1992] Jackson, P. 1992. *Introduction to expert systems*. second edition, Addison-Wesley.

- [John et al 1986] John, C. V. and Maxwell, W. L. 1986. *A system for manufacturing scheduling with interactive computer graphics* IIE Transactions, 298-303.
- [Kelly 1989] Kelly, A. 1989. *Maintenance and its management*. Farnham, UK: Conference Communication.
- [Kanet & Hayya 1982] Kanet, J. J. and Hayya, J. C. 1982. *Priority dispatching with operation due dates in job shop*. Journal of Operations Management, No. 2, 167-176.
- [Klein 1986] Klein, B. 1986. *SIMFACTORY tutorial*. 1986 Winter Simulation Conference Proceedings, 530-542.
- [Lee et al 1993] Lee, J. K., Suh, M. S. and Fox, M. S. 1993. *Contingencies for the design of Scheduling Expert Systems*. Expert Systems With Applications, Vol. 6, 219-230.
- [Lenstra et al 1977] Lenstra, J. K., Rinnooy Kan, A. H. G., and Brucker, P. 1977. *Complexity of machine scheduling problems*. Annals of Discrete Mathematics, Vol. 1, 343-362.
- [Lenz 1985] Lenz, J. E. 1985. *MAST: a simulation program as advanced as the system it studies*. The first International Conference on Simulation in Manufacturing.
- [Holloway & Nelson 1974] Holloway, C. A. and Nelson, R. T. 1974. *job shop scheduling with due dates and variable processing times*. Management Science, Vol. 20, No. 9, 1264-1275.
- [Malhotra & Ritzman 1990] Malhotra, M. K. and Ritzman, L. P., 1990. *Resource flexibility issues in multistage manufacturing*. Decision Sciences, Vol. 21, No. 4, 673-690.
- [Marriott 1994] Marriott, B. 1994. *Production scheduling systems using PRO-VISA*. Winter Simulation Conference Proceedings, 522-526.

- [McCarthy & Liu 1993] McCarthy, B. L., Liu, J. 1993. *Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling*. International Journal of Production Research, Vol. 31, No 1, 57-79.
- [McKey et al 1988] McKey, K. N., Safayeni, F. R. and Buzacott, J. A. 1988. *Job-Shop scheduling theory: What is relevant ?*. Interfaces Vol. 18, 84-90.
- [Mellor 1966] Mellor, P. 1966. *A review of job shop scheduling*. Operational Research Quarterly, Vol. 17, 161-171.
- [Melnyk et al 1985] Melnyk S. A., Carter, P. L., Delts, D. M., Lyth, D. M. 1985. *Shop Floor Control*. Dow Jones-Irwin.
- [Melnyk et al 1986a] Melnyk S. A., Vickery, S. K. and Carter, P. L. 1986. *Scheduling, sequencing and dispatching: alternative perspectives*. Production and Inventory Management, Vol. 27, 58-67.
- [Melnyk et al 1986b] Melnyk S. A. and Carter, P.L. 1986. *Identifying the principles of effective production activity control* Annual International Conference Proceedings - American Production and Inventory Control Society, 227-232
- [Melnyk et al 1987] Melnyk S. A. and Carter, P.L. 1987. *Production Activity Control - a practical guide*. Dow Jones-Irwin, Homewood.
- [Melnyk 1988] Melnyk S. A. 1988. *Production Control: Issues and Challenges*. Intelligent Manufacturing, The Benjamin/Cummings Publishing Company. Inc, 199-231.
- [Melnyk et al 1992] Melnyk, S. A., Denzler, D. R., Fredendall, L. 1992. *Variance control vs dispatching efficiency*. Production and Inventory Management, Vol. 33, No. 3 , 6-13.
- [Melnyk et al 1994] Melnyk, S. A., Tan, K. C, Denzler, D. R. and Fredendall, L. 1994. *Evaluating variance control, order review/release and dispatching: a regression analysis*. International Journal of Production Research, Vol. 32, No. 5, 1045-1061.

- [Muhlemann et al 1982] Muhlemann, A. P., Lockett, A. G. and Farn, C. K., 1982, *Job shop scheduling heuristics and frequency of scheduling*. International Journal of Production Research, Vol. 20, No. 2, 227-241.
- [Nakasuka et al 1992] Nakasuka, S. and Yoshida, T. 1992. *Dynamic scheduling system utilizing machine learning as a knowledge acquisition tool*. International Journal of Production Research, Vol. 30, No. 2, 411-431.
- [Nelson 1967] Nelson, R. T. 1967. *Labor and Machine Limited Production Systems*. Management Science, Vol. 13, No. 9, 648-671.
- [Newman 1988] Newman, P. A. 1988. *Scheduling in CIM System*. Artificial Intelligence - implementation for CIM, 361-402, IFS Ltd, UK.
- [Noronha et al 1991] Noronha, S. J., Sarma, V. . S. 1991. *Knowledge based approaches for scheduling problems: A survey*. IEEE Transactions on Knowledge and Data Engineering, Vol. 3, No. 2, 160-171.
- [O'Grady & Lee 1988] O'Grady, P. and Lee, K. H. 1988. *An intelligent cell control system for automated manufacturing*. International Journal of Production Research, Vol. 26, No. 5 845-861.
- [Ow 1985] Ow, P. S. 1985. *Focused Scheduling in Proportionate Flowshops*. Management Science, Vol. 31, No. 7, 852-869.
- [O'Keefe 1986] O'Keefe, R, 1986. *Simulation and expert systems - A taxonomy and some examples*. Simulation, Vol 46, No. 1, 10-16.
- [Panwalker 1977] Panwalker, S. and Iskender, W. 1977. *A survey of scheduling rules*. Operations Research, Vol. 25, 45-61.
- [Park et al 1995] Park, P. S and Salegna, G. J, 1995. *Load smoothing with feedback in a bottleneck job shop*. International Journal of Production Research, Vol. 33, No. 6, 1549-1568.
- [Peele & Chapman] Peele, T. and Chapman, R. L., 1984. *Determining maintenance manpower requirements*. Plant Engineering, Vol. 38, No. 16, 74-77.

- [Pegden 1985] Pegden, C. 1985. *Introduction to SIMAN*. System Modelling Corporation, PA.
- [Pluym 1990] Pluym, B. V. D. 1990. *Knowledge-based decision-making for job shop scheduling*. International Journal of Computer Integrated Manufacturing, Vol. 3. No 6, 354-363.
- [Prabhu et al 1992] Prabhu, P., Sharit, J and Drury, C. 1992. *Classification of temporal errors in CIM systems: development of a framework for deriving human-centred information requirements*. International Journal of Computer Integrated Manufacturing, Vol. 3, No. 2, 68-80.
- [Prather 1983] Prather, K. L. 1983. *Seven deadly Sins of Production Control*. American Production and Inventory Control Society 26th Annual International Conference Proceedings, New Orleans, LA. 218-220.
- [Pritsker 1974] Pritsker, A. A. B. 1974. *The GASP IV Simulation Language*. John Wiley, New York.
- [Pritsker et al 1979] Pritsker, A. A. B. and Pegden, C. D. 1979. *Introduction to Simulation and SLAM*. John Wiley, New York.
- [Raman & Talbot 1993] Raman, N. and Talbot, F. B. 1993. *The job shop tardiness problem: A decomposition approach*. European Journal of Operational Research, Vol. 69, 187-199.
- [Ranky 1988] Ranky, P. G. 1988. *A real-time, rule-based operation control strategy in CIM environment*. International Journal of Computer Integrated Manufacturing, No. 1, 55-72.
- [Rohleder & Mabert 1988] Rohleder, G. L. and Mabert, V. A. 1988. *An evaluation of order release mechanisms in a job-shop environment*. Decision Sciences, Vol. 19, 167-189.
- [Roy 1993] Roy, R. 1993. *Promoting shop-floor stability with predictive and reactive approaches to scheduling and control*. Logistics Technology International, Sterling Publication, 113-116.

- [Roy & Meikle 1995] Roy, R. and Meikle, S. E. 1995. *Role of discrete event simulation techniques in finite capacity scheduling*. Journal of the Operational Research Society, Vol. 46, No. 11, 1310-1321.
- [Sarin et al 1990] Sarin, S. C. and Salgame, R. R. 1990. *Development of a knowledge based system for dynamic scheduling*. International Journal of Production Research, Vol. 28, No. 8, 1499-1512.
- [Savage & Mikurak 1985] Savage, E. and Mikurak, M. 1985. *The integration of a finite scheduling system into Manufacturing Resource Planning*. In Proc. Autofact, 13.35-13.55.
- [Schonberger 1979] Schonberger, R. J. 1979. *Clearest-Road-Ahead Priorities for Shop Floor Control: Moderating Infinite-Capacity-Loading Unevenness*. Production and Inventory Management. Vol. 20, No. 2, 17-27.
- [Scudder 1987] Scudder, G. D. 1987. *Determining overtime policies for a repair shop*. OMEGA (International Journal of Management Science), Vol. 15, No. 3, 197-206, 1987.
- [Scudder et al 1993] Scudder, G. D., Hoffmann, T. R. and Rohleder, T. R. 1993. *Scheduling with forbidden early shipments: Alternative performance criteria and conditions*. International Journal of Production Research, Vol. 31, No. 10, 2287-2305.
- [Shannon et al 1985] Shannon, R. E, Mayer, R. and Adelsberger, H. 1985. *Expert systems and simulation*. Simulation, 44:6, 275-283.
- [Sharma 1987] Sharma, D. 1987. *Manufacturing strategy: an empirical study*. PhD dissertation, Ohio State University.
- [Sheu & Krajewsk 1994] Sheu, C. and Krajewsk, L, K. 1994. *A decision model for corrective maintenance management*. International Journal of Production Research, Vol. 32, No. 6, 1365-1382.

- [Shivnan & Browne 1986] Shivnan, J and Browne, J. 1986. *AI based simulation of advanced manufacturing systems* Proceedings of the 2nd international conference on simulation in manufacturing, Chicago, IFS, 23-34.
- [Smith 1985] Smith, S. 1985. *The use of multiple problem decompositions in time constrained planning tasks*. International Joint Conference on AI, 1013-1015.
- [Smith et al 1986] Smith, S. F., Fox, M. S. and Ow, P. S., 1986. *Constructing and maintaining detailed production plans: Investigation into the development of Knowledge-based factory scheduling systems*. AI Magazine, Fall.
- [Smith & Ow 1990] Smith, S. F. and Ow, P. S. 1990. *OPIS: An integrated framework for generating and revising factory schedules*. Journal of the Operational Research Society, Vol. 41, No. 6, 539-552.
- [Suresh & Chaudhuri 1993] Suresh, V. and Chaudhuri, D., 1993. *Dynamic scheduling - A survey of research*. International Journal of Production Economics, Vol. 32, 53-63.
- [Szelke et al 1994] Szelke, E. and Markus, G. 1994. *Reactive scheduling - an intelligent supervisor function*. IFIP Transactions B: Computer Applications in Technology, 125-145.
- [Trelevent 1987] Trelevent, M. 1987. *The timing of Labor Transfers in Dual Resource Constrained Systems: 'Push' vs, 'Pull' rules*. Decision Sciences, Vol. 18, No. 1, 73-88.
- [Trelevent 1989] Trelevent, M. 1989. *A review of the dual resource constrained system research*. IIE Transactions, Vol. 21, 279-287.
- [Trybula et al 1985] Trybula, W. J. and Ingalls, R. D. 1985. *Simulation of hybrid automation*. General Electric Company, Electronics Automation application Centre, Charlottesville, Virginia.
- [Umble & Srikanth 1990] Umble, M. M. and Srikanth, M. L. 1990. *Synchronous manufacturing*. Cincinnati, South Western.

- [Vollman et al 1990] Vollman, T. E., Berry, W. L. and Whybark, D. C. 1990. *Manufacturing Planning and Control System*. Richard D. Irwin.
- [Vepsalainen et al 1987] Vepsalainen, A. P. J. and Morton, T. E. 1987. *Priority rules for job shops with weighted tardiness costs*. Management Science, Vol. 33, 1035-1047.
- [Waikar et al 1995] Waikar, A. M., Sarker, B. R. and Lal, A. M. 1995. *Comparative study of some priority dispatching rules under different shop loads*. Production Planning and Control, Vol. 6, No. 4, 301-310.
- [Walker & Miller 1986] Walker, T. C. and Miller, R. K., 1986. *Expert Syetems 1986: An Assessment of Technology and Applications*. SEAI Technical Publications, Madson, 121-128.
- [Wallace 1980] Wallace, T. 1980. *APICS Dictionary*. 4th Edition, Wahington, DC: American Production and Inventory Control Society.
- [White 1990] White, Jr. K. P. 1990. *Advances in the theory and practice of production scheduling*. Control and Dynamic Systems, Vol. 37, 115-157.
- [Wilbrecht & Prescott 1969] Wilbrecht, J. K. and Prescott, W. R. 1969. *The influence of setup time on job shop performance*. Management Science, Vol. 16, B-274.
- [Wild 1971] Wild, R. 1971. *The Techniques of Production Management*. Holt, Rinehart and Winton, New York.
- [Wortman & Miner 1986] Wortman, D. B. and Miner, R. J. 1986. *Designing flexible manufacturing systems using simulation*. Proceedings of AUTOFACT, Vol. 6, 24-29.
- [Wu & Wysk 1989] Wu, SY. D and Wysk, R. A. 1989. *An application of discrete-event simulation to on-line control and scheduling in flexible manufacturing*. International Journal of Production Research, Vol. 27, No. 9, 1603-1623.
- [Zeestraten 1990] Zeestraten, M. J. 1990. *The look ahead dispatching procedure*. International Journal of Production Research, Vol. 28, No. 2, 369-384.